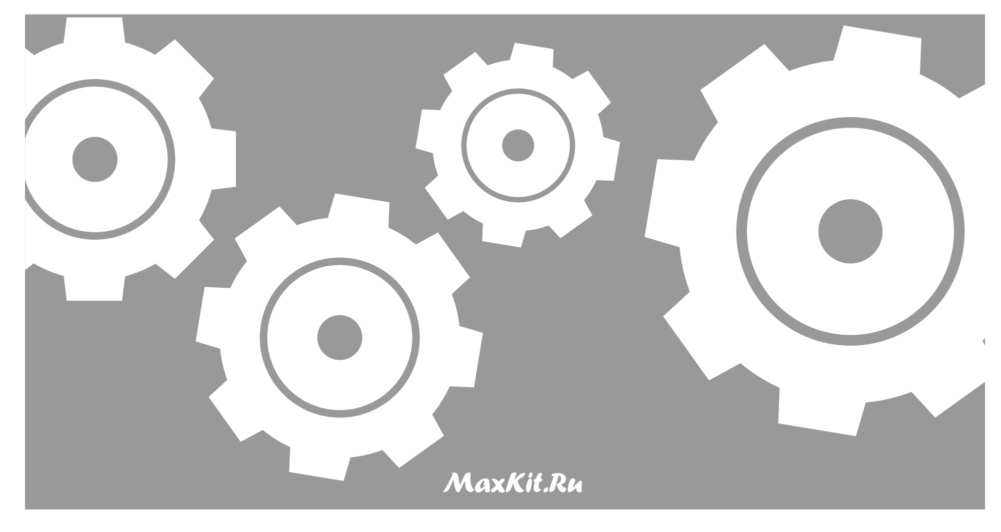
БЫСТРЫЙ СТАРТ

ПЕРВЫЕ ШАГИ ПО ОСВОЕНИЮ ARDUINO

НАБОР КОНСТРУКТОР НАЧИНАЮЩЕГО ИЗОБРЕТАТЕЛЯ



Оглавление



Быстрый старт с набором Arduino

Стартовый набор-конструктор с платой Arduino - Ваш пропуск в мир программирования, конструирования и электронного творчества. Эта брошюра содержит всю информацию для ознакомления с платой Arduino, а также 14 практических экспериментов с применением различных электронных компонентов и модулей. Полученные знания, в дальнейшем, дадут возможность создавать свои собственные проекты и с легкостью воплощать их в жизнь. За дело!

http://maxkit.ru

Часть 1: *Быстрый старт - Arduino*

Часть 2: *Быстрый старт - Электроника*

Что такое Arduino?



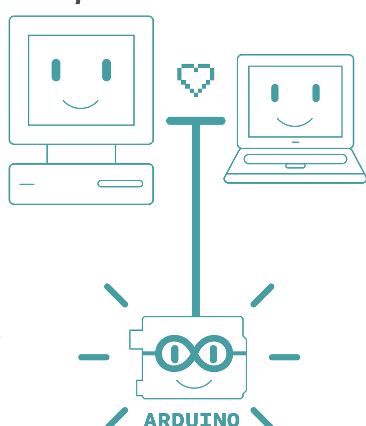
Время Arduino

Arduino — это электронный конструктор и удобная платформа быстрой разработки электронных устройств для новичков и профессионалов. Платформа стала популярной во всем мире благодаря удобству языка программирования, а также открытой архитектуре и программному коду. Программируется через USB без использования программаторов.

Домашняя страничка http://arduino.cc

Компьютер для взаимодействия с миром.

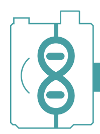
Симпатичная сине-зеленая плата в Вашей руке, или на столе это и есть Arduino. По своей сути Arduino это маленький персональный компьютер, который позволяет выйти за рамки виртуального мира в физический и взаимодействовать с ним. Устройства на базе Arduino могут получать информацию об окружающей среде посредством различных датчиков, а также могут управлять различными исполнительными устройствами.

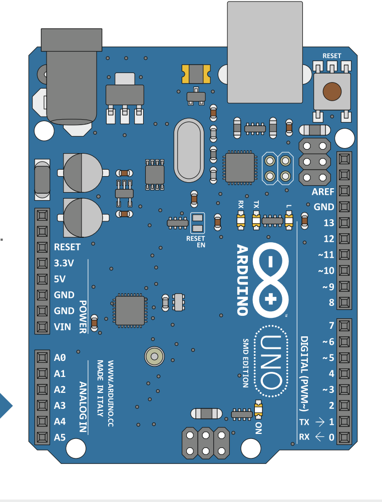


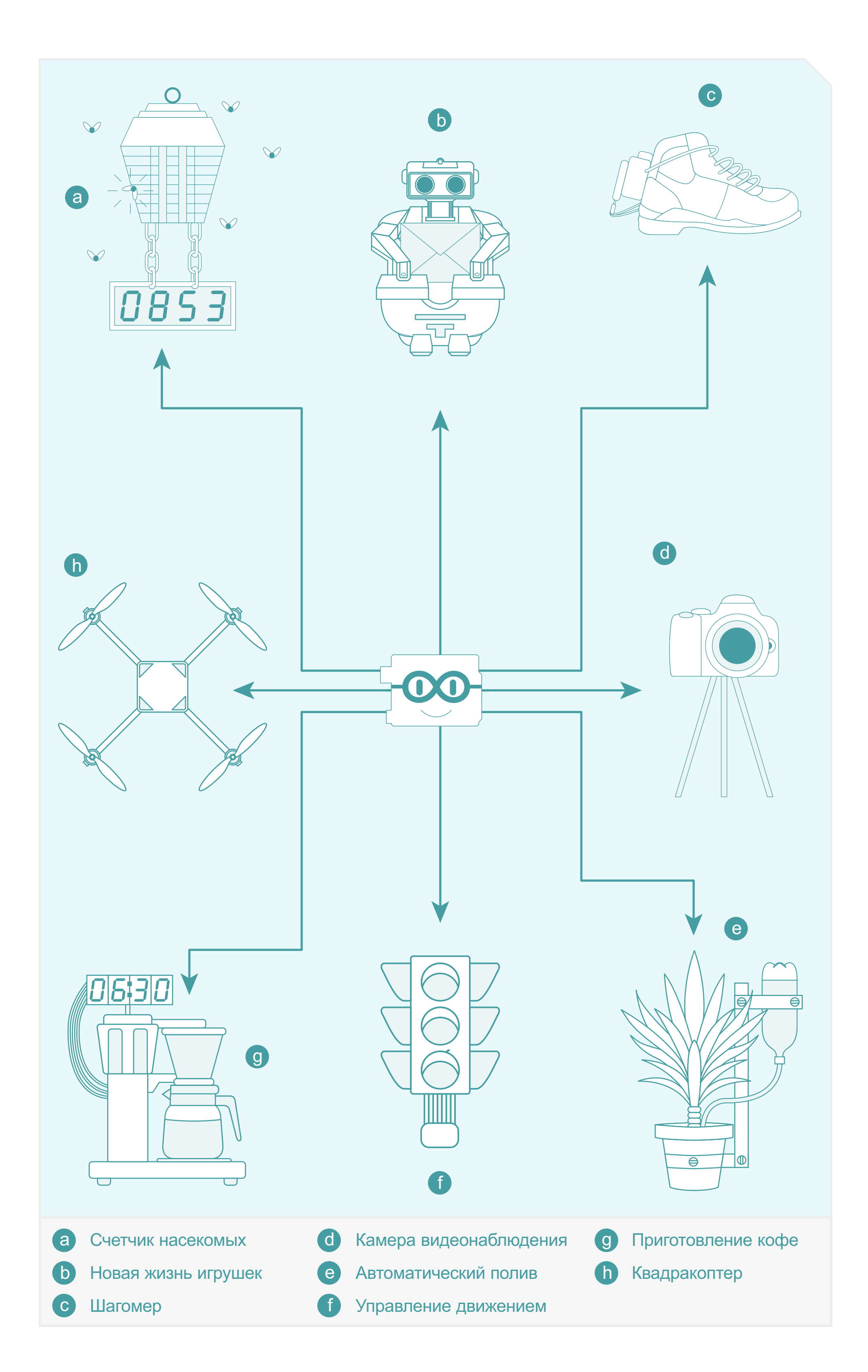
Проекты устройств, основанные на Arduino, могут работать самостоятельно, либо взаимодействовать с программным обеспечением компьютера. Попробуйте воспользоваться нашими советами, и вскоре все это обретет реальный смысл.

// Arduino UNO R3, или Arduino UNO R3 SMD

Arduino UNO - плата разработчиков, одна из нескольких (Mega, Nano и пр.), основана на микроконтроллере ATmega328. Он популярен, главным образом, из-за обширной сети поддержки и универсальности.
У Arduino UNO есть 14 цифровых портов ввода - вывода, шесть из которых умеют выдавать ШИМ. Еще есть 6 входящих аналоговых портов. Есть генератор 16 Мгц, USB порт, разъем питания, кнопка сброса, и разъем ICSP.






a
0853
b
c
h
d
06:30
g
e
f
a Счетчик насекомых
b Новая жизнь игрушек
c Шагомер
d Камера видеонаблюдения
e Автоматический полив
f Управление движением
g Приготовление кофе
h Квадракоптер

Скачайте среду разработки Arduino IDE



Доступ в интернет

Для того, чтобы начать работу с Arduino вам необходимо зайти на сайт arduino.cc и скачать программу Arduino IDE из раздела Download, совершенно бесплатно. Предварительно уточните, у знающих людей, тип операционной системы на вашем компьютере.

С помощью любого браузера, внести следующее в адресную сроку:

arduino.cc/en/Main/Software

1

Для загрузки

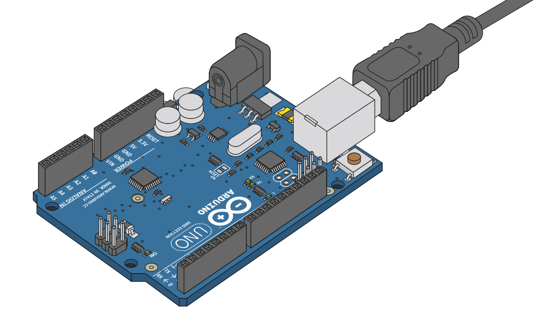
Выберите программу, подходящую под вашу операционную систему

- **Windows Installer (.exe), Windows (ZIP file)**
- **Mac OS X**
- **Linux: 32 bit, 64 bit**
- **source**

❗ Обязательно выберите программу, подходящую под вашу операционную систему!

// Подсоедините вашу плату Arduino Uno к компьютеру

Используйте USB кабель, входящий в комплект набора, соедините плату Arduino с одним из USB разъемов вашего компьютера.



2

3

// Установка драйверов

В зависимости от того, какая операционная система установлена на вашем компьютере, вам необходимо выбрать способ установки драйвера.
Ниже даны ссылки и методы установки для платы Arduino Uno.

*** Вы должны очень внимательно отнестись к выбору и установке драйверов.**



Установка для Windows

Пройдите по ссылке ниже для доступа к инструкциям по установке на компьютер с операционной системой Windows.

http://arduino.cc/en/Guide/Windows



Установка для Macintosh OS X

Для Mac OS не требуется устанавливать дополнительные драйвера, но вы можете посетить страничку “быстрый старт”.

http://arduino.cc/en/Guide/MacOSX



Установка для Linux: 32 bit / 64 bit

Пройдите по ссылке ниже для доступа к инструкциям по установке на компьютер с операционной системой Linux.

http://www.arduino.cc/playground/Learning/Linux



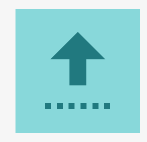
// Интерфейс программы Arduino IDE:

Так выглядит среда программирования Arduino IDE, запущенная на вашем компьютере. Мы начинаем знакомство с её интерфейсом. Сейчас мы не будем касаться кода, а расскажем лишь о назначении функциональных клавиш в меню программы.

GUI (Graphical User Interface - Графический интерфейс пользователя)

1. **Verify (Проверить):** Compiles (компилирует, собирает) проект и проверяет на отсутствие ошибок в коде программы.
2. **Upload (Загрузить):** Загрузить программу в микроконтроллер на плате Arduino. Убедитесь в подключении платы.
3. **New (Новый):** Создать новую программу, скетч.
4. **Open (Открыть):** Открывает меню со списком проектов, скетчей.
5. **Save (Сохранить):** Сохраняет активный, текущий проект.
6. **Serial Monitor (Сериал Монитор):** Отображает работу COM порта с текущим скетчем.
7. **Sketch Name (Имя скетча):** Отображается имя текущего проекта (скетча).
8. **Code Area(Оласть Кода):** Область кода программы (скетча).
9. **Message Area (Область Сообщений):** Область сообщений. Также сюда выводятся сообщения об ошибках.

// Ниже Вы видите три самых важных команды в меню Arduino IDE:



Open (Открыть)



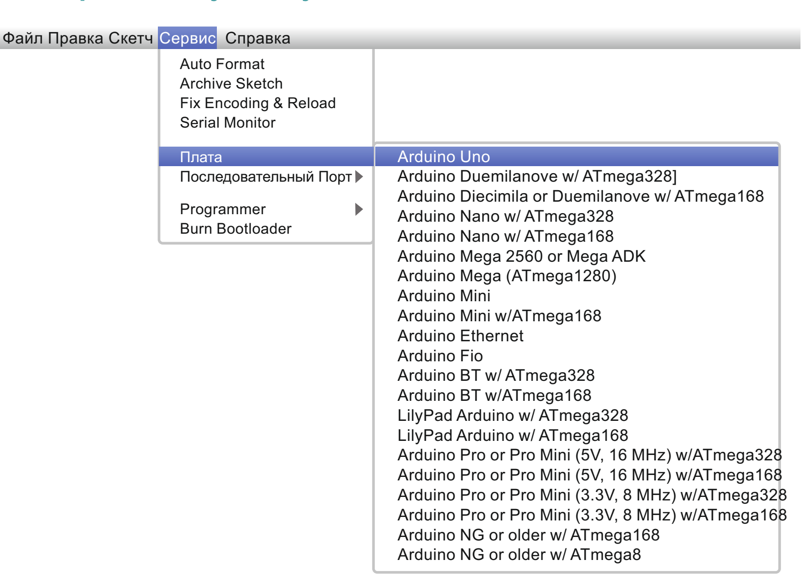
Verify (Проверить)



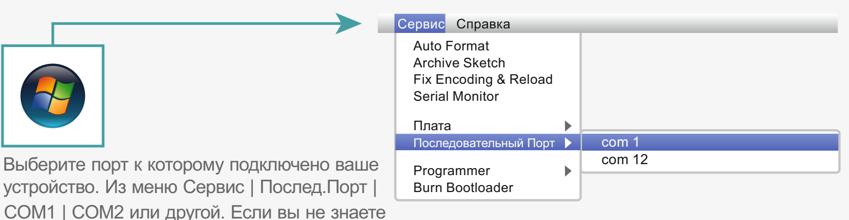
Upload (Загрузить)



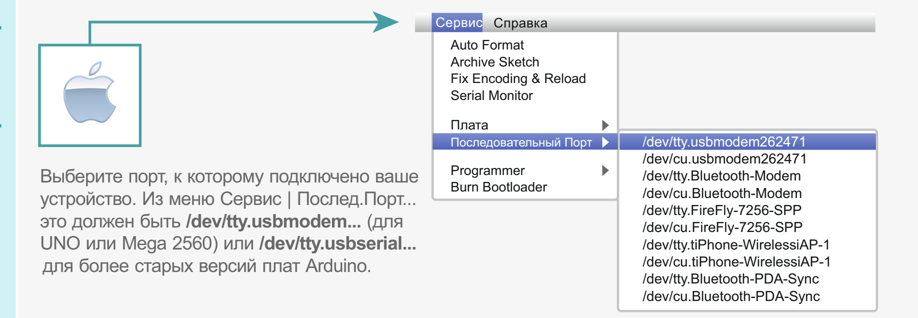
// Выберите вашу плату: *Arduino Uno*



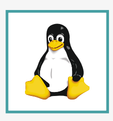
// Выберите порт подключения устройства



Выберите порт к которому подключено ваше устройство. Из меню Сервис | Послед.Порт | COM1 | COM2 или другой. Если вы не знаете к какому порту подключена плата, сделайте следующее: Откройте «Панель управления», «Диспетчер устройств», раскройте пункт «Контроллеры USB», затем отключите разъем USB от компьютера и вы увидите, что одна из надписей пропадет. Если меню свернется, раскройте его заново, запомните все присутствующие порты, затем снова подключите USB от Arduino к компьютеру. Снова раскройте меню, новый появившийся порт будет искомым портом подключения. Выделите его в меню Arduino.



Выберите порт, к которому подключено ваше устройство. Из меню Сервис | Послед.Порт... это должен быть **/dev/tty.usbmodem...** (для UNO или Mega 2560) или **/dev/tty.usbserial...** для более старых версий плат Arduino.



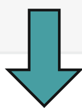
http://www.arduino.cc/playground/Learning/Linux

Скачайте скетчи Arduino (Коды программ используемые в данном руководстве)

5

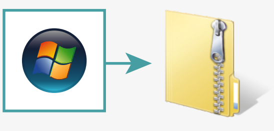


Воспользуйтесь следующей ссылкой, чтобы скачать исходные тексты к опытам:



maxkit.ru/kitcode01

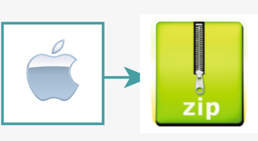
// Скопируйте “MaxKit Code” в папку библиотек “Examples” куда установлена Arduino IDE



Распакуйте MaxKit.rar.
По умолчанию скачанный архив находится в папке “Загрузки”.
Щелкните правой кнопкой мыши по скачанному файлу и выберите распаковать.



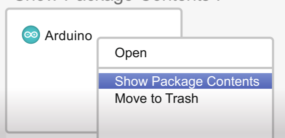
Копируйте папку с распакованными файлами “MaxKit” в папку сназванием “examples”.

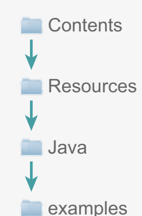


Распакуйте MaxKit.rar.
По умолчанию скачанный архив находится в папке “Загрузки”
Щелкните правой кнопкой мыши по скачанному файлу и выберите распаковать.

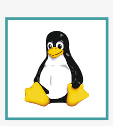


Найдите папку с установленной Arduino. Правый щелчек по ней (ctrl+click) и выберите “Show Package Contents”.



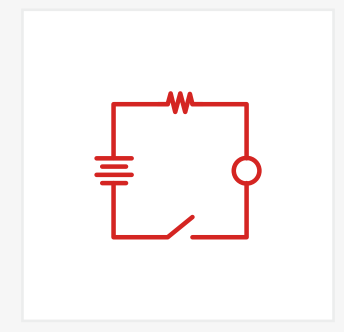


Копируйте папку MaxKit” в папку с установленной Arduino IDE “examples”.



http://www.arduino.cc/playground/Learning/Linux

Приступая к работе с электрическими схемами



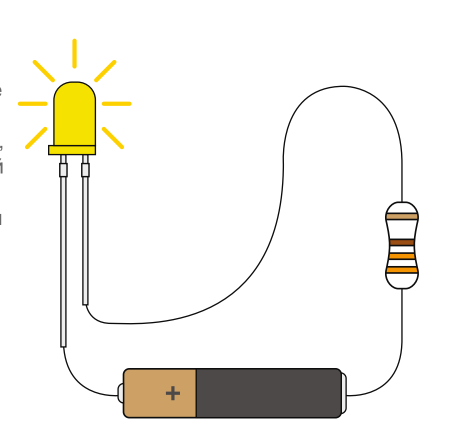
Что такое электрическая цепь?

Электрическая цепь — это неразрывный путь, по которому электрический ток может течь от начальной точки к конечной, например от плюса к минусу. Электрические цепи могут содержать резисторы, диоды, транзисторы, моторы и многое другое. Схема может содержать огромное количество таких элементов.

Схемы, как правило, делятся на три категории: аналоговые схемы, цифровые схемы, и схемы со смешанными сигналами. В этом руководстве вы будете исследовать все три набора схем.

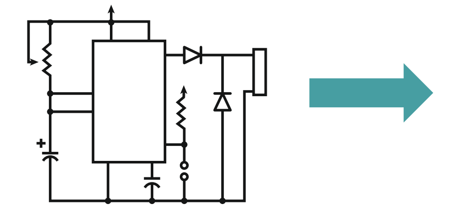
Использование электрических цепей

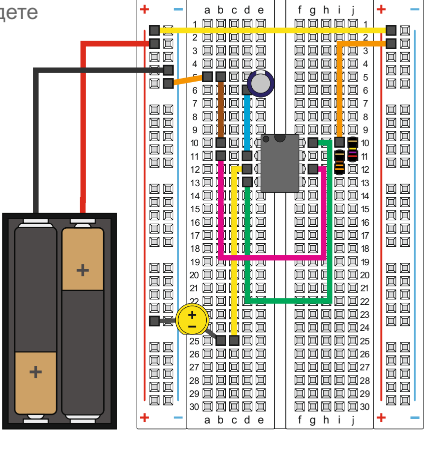
Куда бы вы не посмотрели, везде вы найдете электронные схемы. Телефон в кармане, компьютер, ноутбук, ваша игровая приставка, все эти вещи состоят из электрических цепей и электронных схем. В этом руководстве, вы будете экспериментировать с некоторыми простыми схемами и узнаете как работают некоторые электронные компоненты.

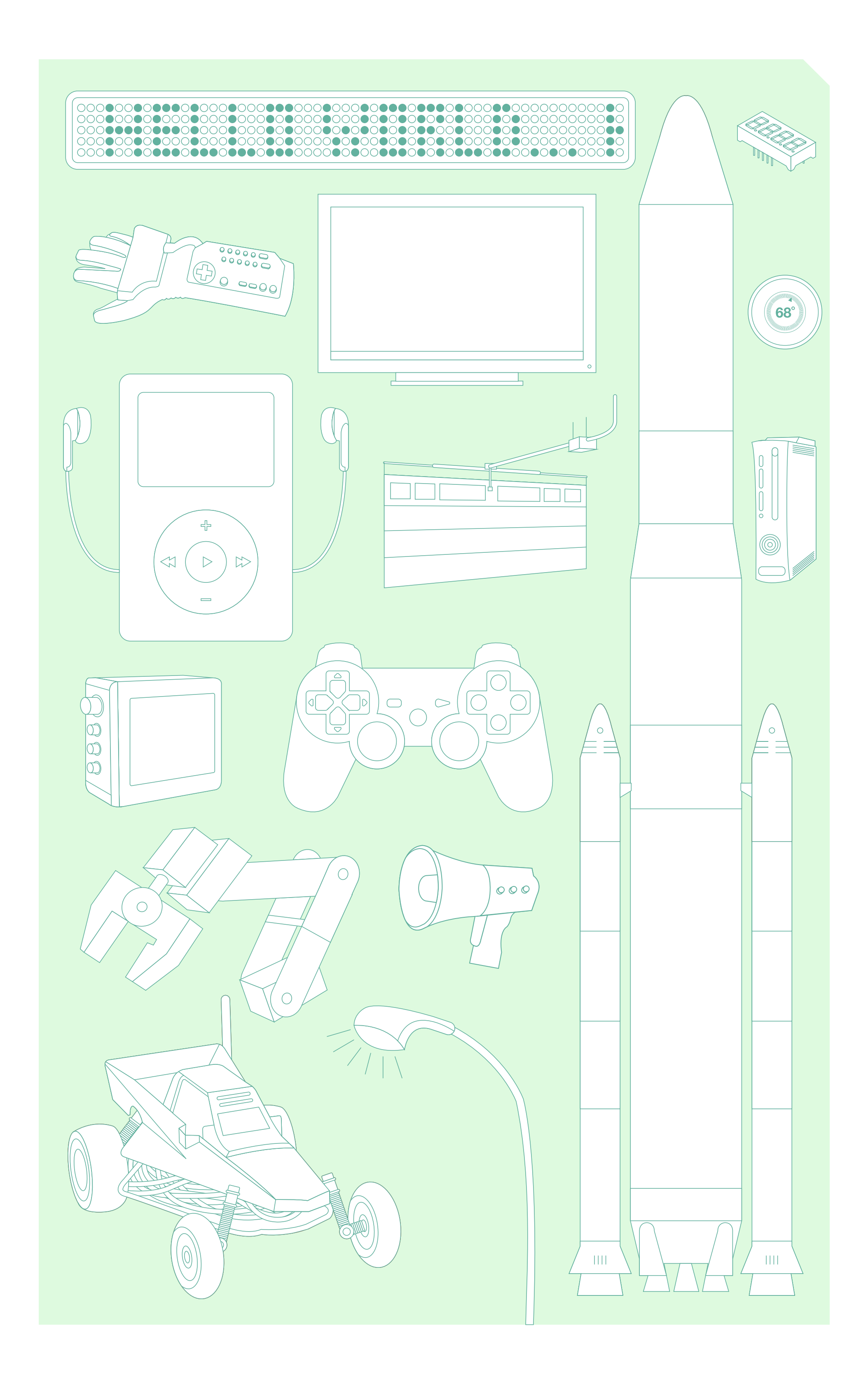


// Простые и сложные схемы

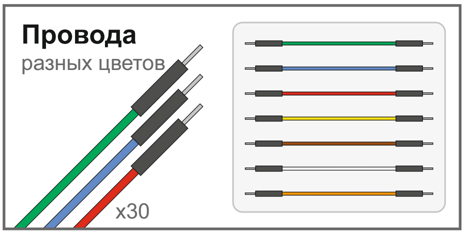
В этом руководстве, в первую очередь, вы будете изучать простые схемы - но это не значит, что вы не сможете сделать что нибудь сложное с помощью простых инструментов! Все, даже самые сложные схемы, состоят из простых элементов. Когда вы закончите изучать это руководство, ваше знание схем позволит вам создавать свои собственные, удивительные по красоте проекты.

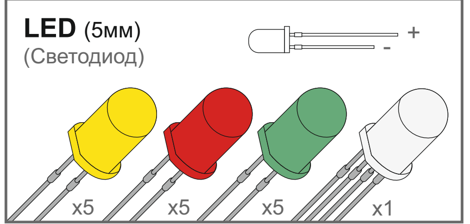


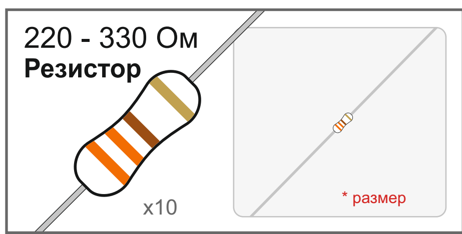


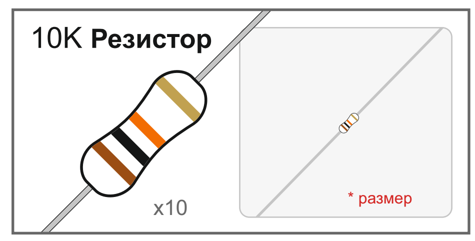

68°

Основные комплектующие

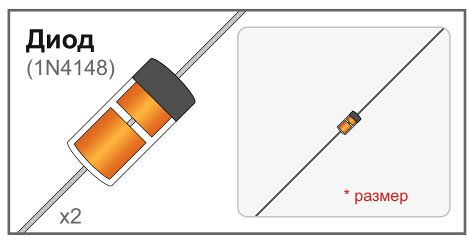




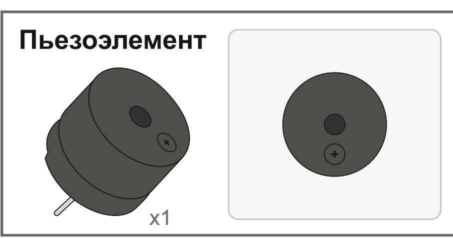


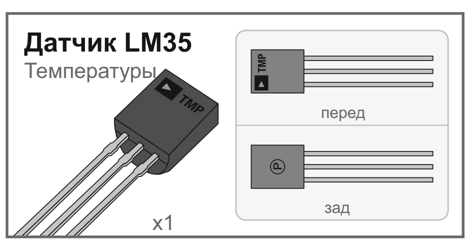


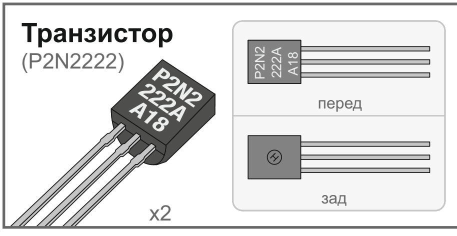


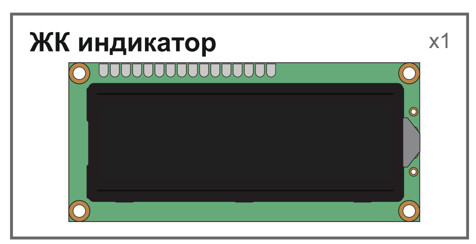


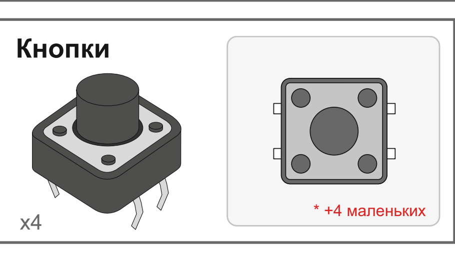






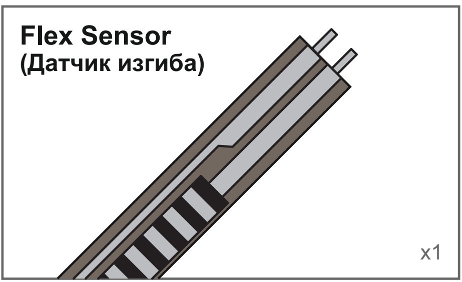






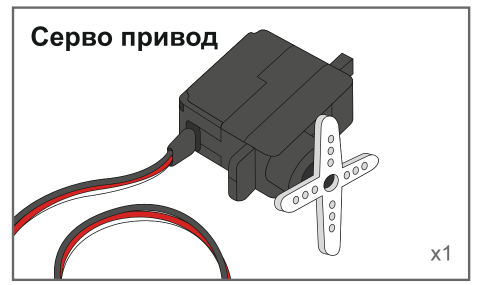
Основная комплектация - продолжение

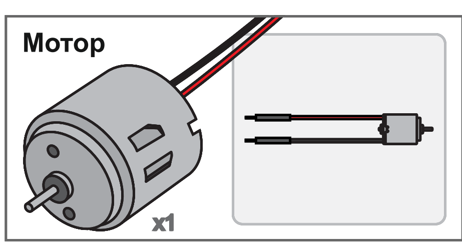


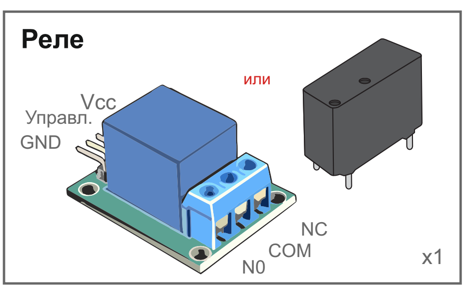


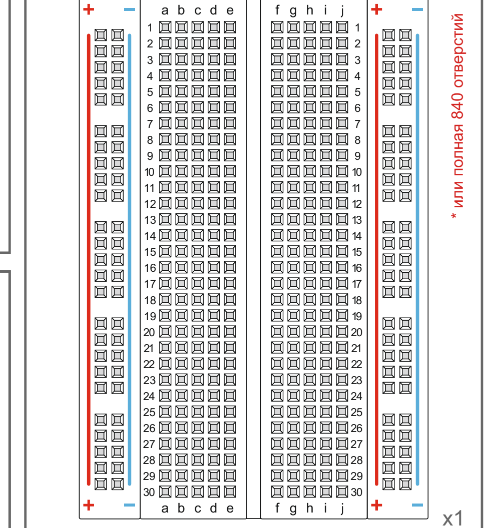






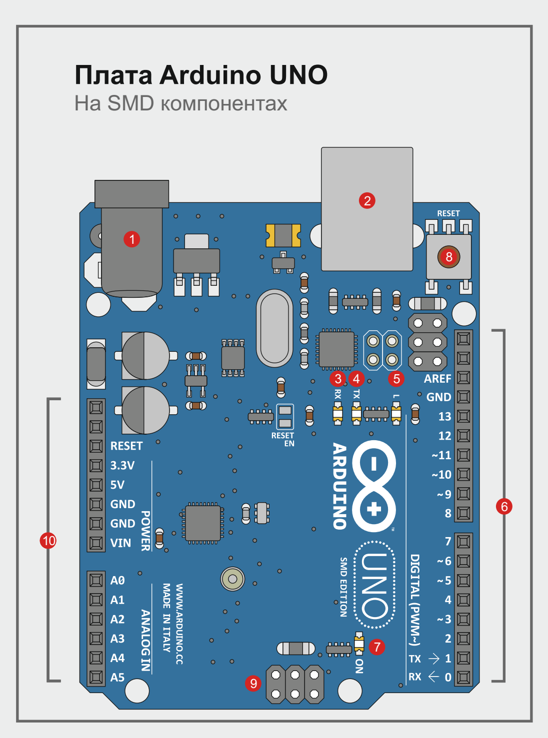


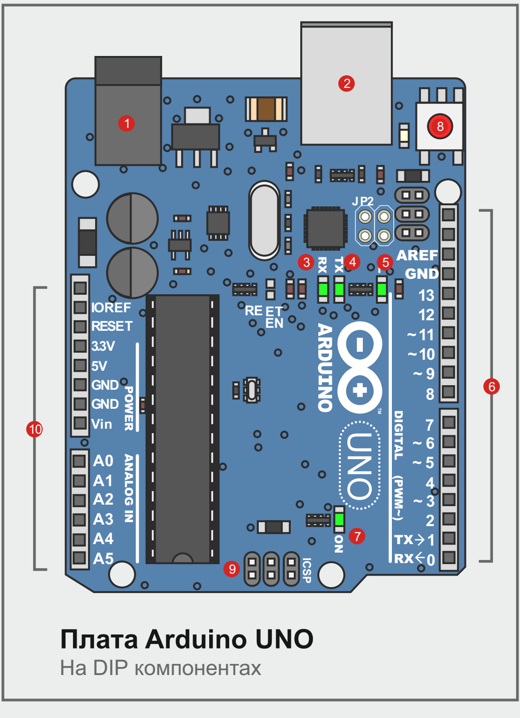




Плата Arduino UNO

На SMD компонентах





Плата Arduino UNO

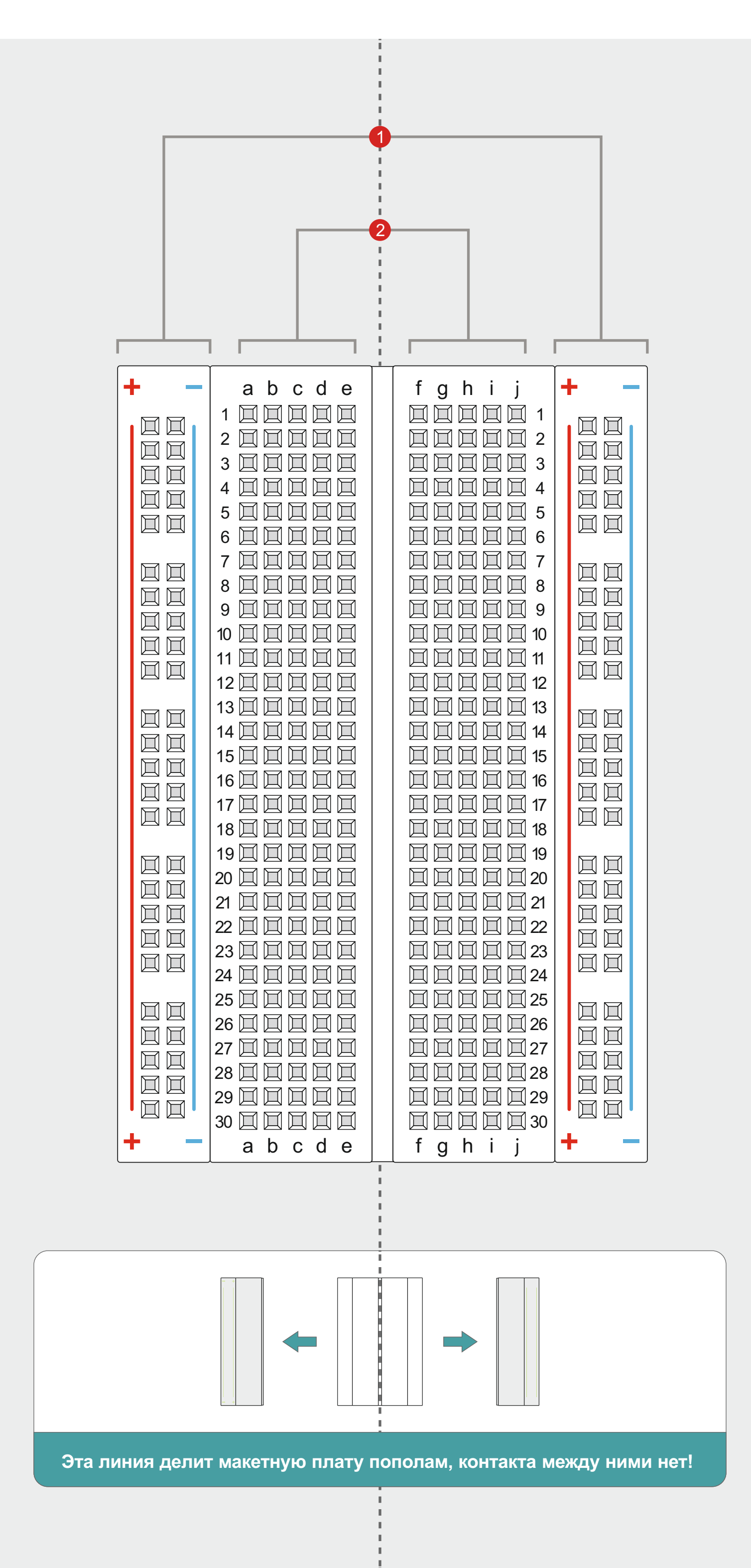
На DIP компонентах

Arduino Uno

1. **Разъем Питания** (от батареи) - Может использоваться с блоками питания 9 - 12 Вольт.
2. **Разъем USB** (USB Порт) - Может использоваться для питания схем, а также для связи с компьютером.
3. **Индикатор** (RX: Прием) - Используется для индикации приема данных, если конечно это прописано в программе.
4. **Индикатор** (TX: Передача) - Используется для индикации передачи данных, если конечно это прописано в программе.
5. **Индикатор** (Порт 13: поиск неисправностей) - Во время работы скетча показывает правильно ли всё работает.
6. **Порты** (ARef, Ground, Digital, Rx, Tx) - опорное напряжение, земля, цифровые порты, порты приема и передачи данных.
7. **Индикатор** (Индикатор питания) - сигнализирует о подаче питания на плату Arduino.
8. **Reset** (сброс) - ручной перезапуск платы Arduino, приводит к перезапуску вашей программы.
9. **Разъем ICSP** (Порт программирования) - дает возможность программировать, минуя загрузчик самой платы.
10. **Порты** (Analog In, Power In, Ground, Power Out, Reset) - аналоговые, входящие, исходящие, питание и земля.

// Диаграмма Портов

Pins, или «Порты», расположенные на плате Arduino являются наиболее важной частью создаваемых вами электронных схем. Найдите минутку и внимательно изучите порты ввода/вывода.


1
2
+ –
a b c d e
f g h i j
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
Эта линия делит макетную плату пополам, контакта между ними нет!

Макетная плата

1 Вертикальные соединения (+ Питания и Земля (GND))

2 Горизонтальные соединения (a-e & f-j)

Как происходят соединения?

Установление Соединения:

СВЕТОДИОД

ПОДСОЕДИНЕН!

внутри макетной платы

+ Питание:

Каждый элемент соединен с “плюсом” в любом месте вертикального столбца.

– Земля:

Каждый элемент соединен с “землей” в любом месте вертикального столбца.

Горизонтальные ряды:

Каждая строка с 1 по 30 состоят из 5 соединенных между собой ячеек. Электронные компоненты подсоединенные в ряд из пяти ячеек будут включенны в одну электронную цепь.

Вид изнутри >>>

ОПЫТ № 1 - *Ваша первая схема*

Ваши действия:

1 СОБЕРИТЕ | **2 ЗАПИШИТЕ** | **3 ЗАГРУЗИТЕ**

+ Для того, чтобы хватило длины проводов, не располагайте платы далеко друг от друга.

+ Соедините Arduino и компьютер кабелем USB из вашего набора

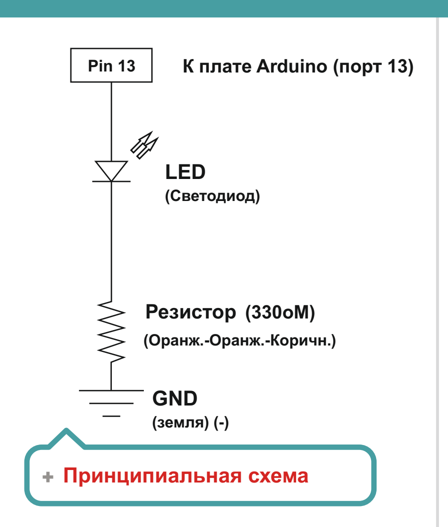
+ Из этого места вы берете +5 вольт и GND для питания макетной платы.



5V - (Пять вольт) это рабочее напряжение платы Arduino!
Это напряжение будет поступать с вашего компьютера через порт USB и будет питать все компоненты ваших схем, а также саму плату Arduino. Это напряжение вы не можете почувствовать, и оно никак не может вам навредить.

1

Мигающий светодиод



+ **Принципиальная схема**

LEDs (light-emitting diodes) по-русски **светодиод,** он используется во многих устройствах. Мы начнем наши опыты с самой простой схемы - “Arduino и мигающий светодиод”.
Начнем с того, что попробуем заставить Arduino помигать нам приветственно светодиодом, да, да - именно заставить, потому что мы обладаем всей полнотой власти над этой маленькой, но очень мощной платой, под названием Arduino. Подберем необходимые детали и приступим...

+ **В этой секции располагается название, краткое описание опыта, а также что должно получиться в результате.**

ДЕТАЛИ:

LED x1
330Ω Резистор x1
Провод x3

+ **В этой секции указываются детали необходимые к опыту**

+ **На этом рисунке показано как удобнее располагать платы и соединительные провода относительно друг друга. Но Вы можете располагать их как Вам удобно.**

+ **Для удобства использования таких деталей как резисторы, им желательно согнуть выводы под углом 90 градусов.**

Опыт 1: Мигающий светодиод

Компоненты	Обозначение	Arduino	Макетная плата	
Светодиод 5мм			c2 (+) - c3 (-)	Ножки светодиода вставляются в макетную плату c2(длинная ножка), c3(короткая ножка).
330Ω Резистор			a3 - -	Один вывод резистора вставляется в контакт a3, а другой в общий “-” макетной платы.
Провод перемычка		GND	-	“GND” с платы Arduino, соединяется с общим “-” на макетной плате.
Провод перемычка		5V	+	“5V” с платы Arduino, соединяется с общим “+” на макетной плате.
Провод перемычка		Pin 13	e2	“Pin13” с платы Arduino соединяется с контактом ”e2” на макетной плате.

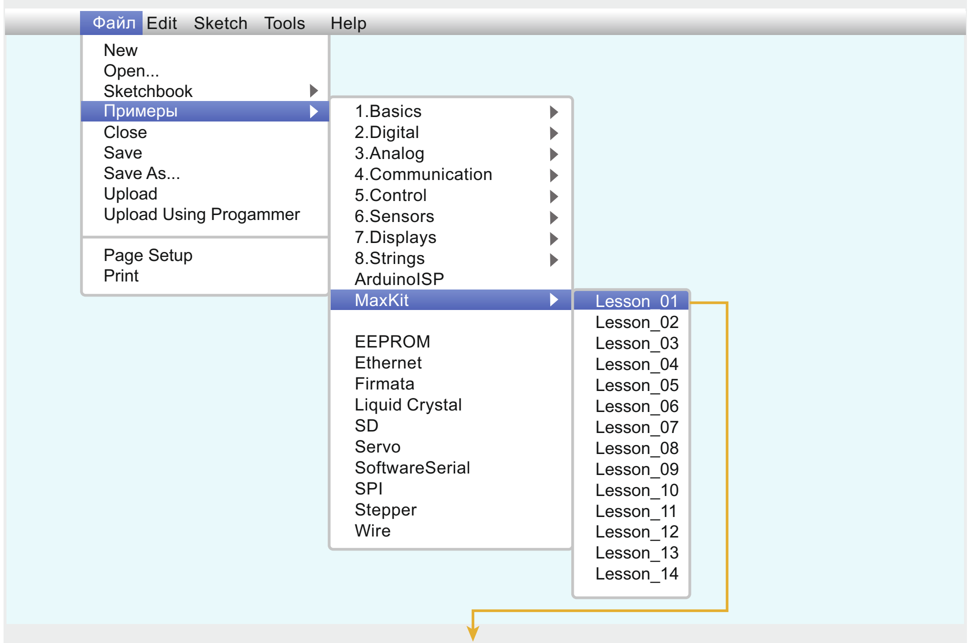
Arduino: В столбике с синим фоном указанны порты, или пины на плате Arduino

Макетная плата: В столбике с белым фоном указанны контакты на макетной плате



Откройте ваш первый скетч:

Запустите среду программирования Arduino IDE на вашем компьютере. В настройках программы можете выбрать ваш родной язык интерфейса. Откройте скетч с первым опытом из меню Файл, Примеры, MaxKit”:



// Опыт #1

Lesson_01

```
/*
 * Набор для экспериментов MaxKit 01
 * Код программы для опыта №1: sketch 01
 *
 * МИГАЮЩИЙ СВЕТОДИОД
 *
 *
 *
 */


void setup() {


 pinMode(13, OUTPUT);  //Настроим pin 13, как исходящий.
}

void loop() {
 digitalWrite(13, HIGH);   // Включить светодиод
 delay(1000);              // Ожидание - 1 секунда
 digitalWrite(13, LOW);    // Выключить светодиод
 delay(1000);              // Ожидание - 1 секунда
}
```



Проверка

после того, как программа написана, необходимо дать компьютеру ее проверить, и если всё правильно, можно переходить к следующему шагу.



Загрузка

кнопка «Загрузка» посылает скомпилированную программу в плату Arduino через USB шнур, после полной загрузки она сразу начнет свою работу.

// результат выполнения программы.

1



Откройте Arduino IDE // Файл > Примеры > MaxKit > lesson_1

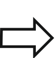
+ Здесь указан номер скетча используемого в данном опыте

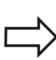
На заметку:

+ Не забудьте внимательно проверить и загрузить код.



+ Для того, чтобы понять как работает, код смотри ниже

`pinMode(13, OUTPUT);`  Прежде чем использовать один из портов, вы должны сообщить Arduino это INPUT(входящий) или OUTPUT(исходящий), для этого мы используем функцию pinMode().

`digitalWrite(13, HIGH);`  Прежде чем использовать порт в качестве ИСХОДЯЩЕГО, вы должны подать команду HIGH (5 Вольт), или LOW (0 Вольт).

Что вы должны увидеть:

+ Здесь описано что должно происходить со схемой в итоге.

Вы увидите мигающий светодиод, если этого не произошло, убедитесь в правильности собранной схемы, а также загружен ли код программы в Arduino. Если все же программа не выполняется, смотрите секцию по устранению ошибок, ниже.

+ В этой секции описываются возможные трудности или ошибки, которые могут возникать во время эксперимента.

+ Здесь вы найдете примеры из жизни, где могут использоваться созданные Вами схемы. На самом деле многие из схем используются вами каждый день.

Возможные трудности:

Светодиод не светится?
Светодиоды имеют полярность - плюс и минус, попробуйте повернуть его на 180 градусов. Не беспокойтесь за его исправность.

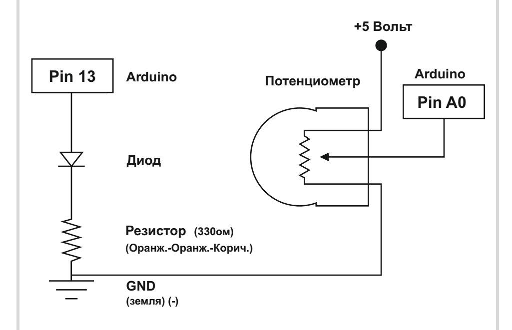
Программа не загружается.
Такое иногда случается из-за ошибочного указанного порта подключения к компьютеру, смотрите Сервис > Последовательный порт >

Все еще не работает.
Скорее всего поломана схема, это уже не шутки, свяжитесь с нами по почте, мы попробуем помочь!

Применение в жизни:

Почти во всех современных телевизорах и мониторах есть светодиодные индикаторы, которые показывают включен прибор или нет.

ОПЫТ #2



2 Потенциометр

В этом опыте вы будете работать с потенциометром. Потенциометр можно назвать переменным резистором. Когда потенциометр одним из своих контактов, нижним по схеме, соединен с землей (минус питания), а другим (верхним) к +5 вольт, то на среднем контакте мы получим напряжение от 0 до 5 вольт, в зависимости от положения ручки потенциометра.
Потенциометры обычно используют для регулировки различных параметров, например громкости или яркости. Из этого опыта вы узнаете, как использовать потенциометр для управления яркостью светодиода.

ДЕТАЛИ:

Потенциометр	Светодиод	330Ω Резистор	Провод
x1	x1	x1	x6

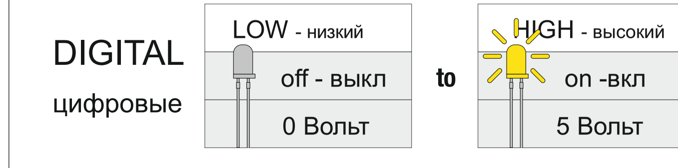
Опыт 2: Потенциометр

Компоненты	Обозначение		
Потенциометр			a6 - a7 - a8
Светодиод 5мм	+ -		h20 (+) - h21 (-)
330Ω Резистор			i21 - -
Провод перемычка			e6 - -
Провод перемычка		A0	e7
Провод перемычка			e8 - +
Провод перемычка		Pin 13	j20
Провод перемычка		5V	+
Провод перемычка		GND	-

Цифровые против аналоговых:

Если вы взглянете на вашу Arduino вы увидите надписи "DIGITAL", или "ANALOG". Что это такое и для чего?

Для многих устройств, таких например как кнопки переключатели имеют всего два рабочих состояния - Вкл., Выкл, или если говорить языком Arduino "HIGH" (5 Вольт - вкл.) и "LOW" (0 Вольт выкл.). Эти значения для цифровых портов и для аналоговых понимаются Arduino по разному.
Цифровые порты Arduino сильны в обработке и передаче этих сигналов они могут включать, выключать, кодировать, и пр. пр.



Но существует множество приборов и компонентов для которых существуют состояния не только Вкл. и Выкл. Примером таких являются различные датчики температур, влажности и пр.
Для таких устройств у Arduino есть шесть аналоговых портов, которые переводят входное напряжение в числовой диапазон от 0 - 0 вольт до 1023 5 вольт. Это идеальное решение для измерений.





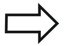
Код Arduino:



Откройте Arduino IDE // Файл > Примеры > MaxKit > lesson_2

На заметку:

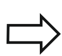


`int sensorValue;` 

Переменная «sensorValue» хранит в себе числовое значение. Необходимо ввести или объявить переменную до начала ее использования. Здесь мы дали имя переменной sensorValue (чувствительно к регистру) и объявили ее «int» целым числом.

`= analogRead(sensorPin);` 

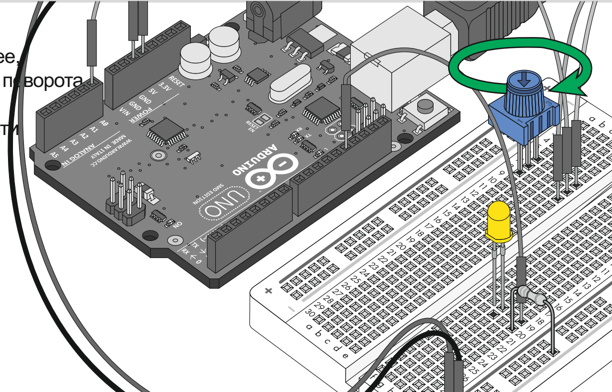
Функция analogRead(), считывает значение с указанного аналогового входа примерно 10,000 раз/сек. Напряжение от 0 до 5 вольт преобразует Аналогово Цифровой Преобразователь в значение от 0 до 1023, шаг 0.0049 Вольт.

`delay(sensorValue);` 

Arduino очень быстро работает, и может обрабатывать тысячи строк кода за секунду. Для того, чтобы рассмотреть что в данный момент происходит, нужно ввести некую задержку. Delay() - пауза, принимает значение 1000 ms - 1 секунда.

Что вы должны увидеть:

Вы должны увидеть мигающий светодиод, который мигает быстрее, или медленнее, в зависимости от поворота ручки потенциометра. Если он не работает, убедитесь в правильности собранной схемы и в правильности загруженного кода.



Возможные трудности:

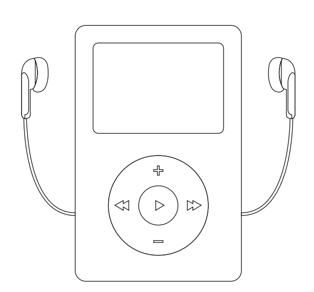
Неравномерная работа
Из за ненадежного контакта ножек потенциометра. Проверьте надежность слегка придавив его в макетную плату.

Не работает
Убедитесь, что вы случайно не соединили среднюю ножку потенциометра с цифровым портом 2, вместо аналогового. А также Плюс питания с землей.

Вернуться к началу
Вы можете попробовать все разобрать и собрать заново, иногда это помогает.

Применение в жизни:

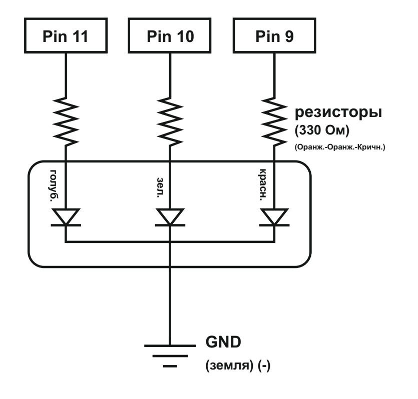
MP3 проигрыватель - регулятор громкости есть не что иное как потенциометр.



ОПЫТ #3

3

RGB - светодиод



Что может быть прекраснее мигающего светодиода? Трехцветный светодиод. RGB - Red, Green, Blue! Красный, Зеленый и Синий - три в одном! Три светодиода в одном корпусе.
Они могут удивительно переливаться, выдавая неповторимые оттенки и световые эффекты, это зависит от того, на сколько ярко горит тот, или иной светодиод.
В этом опыте вы научитесь подключать его к Arduino, а так же как им управлять.

детали:

Свето-диод x1	330Ω Резистор x3	Провод x6

голубой
красный
общий
зеленый

Опыт 3: Трехцветный светодиод

красный
общий (земля)
зеленый
голубой

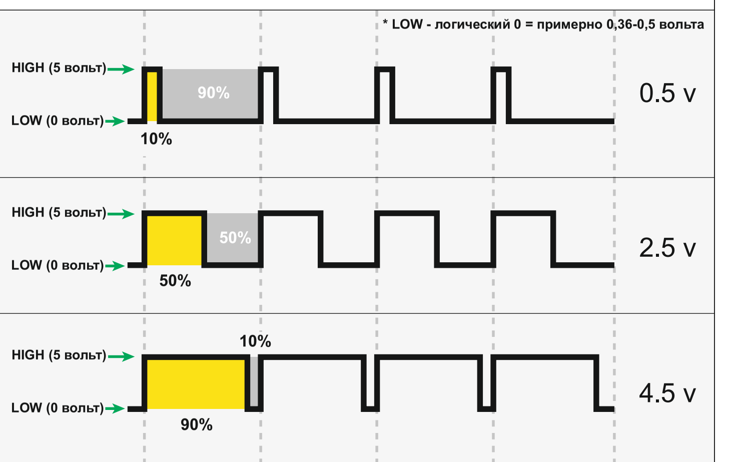
* Самая длинная ножка - общий для всех (gnd)

Компоненты	Обозначение		
RGB Светодиод			a4 a5 a6 a7
330Ω Резистор			e4 - g4
330Ω Резистор			e6 - g6
330Ω Резистор			e7 - g7
Провод перемычка		Pin 9	h4
Провод перемычка			e5 - -
Провод перемычка		Pin 10	h6
Провод перемычка		Pin 11	h7
Провод перемычка		5V	+
Провод перемычка		GND	-

Шокирующая правда про analogWrite():

Мы помним, что Arduino может работать с аналоговым входным напряжением (от 0 до 5 Вольт), используя функцию **analogRead().** А сможет ли она справиться с аналоговым **выходным** напряжением?
Ответ и да и нет. Выдавать на аналоговый порт 5V не умеет, но может притворяться, что умеет. Для этого она использует ШИМ - Импульсную Модуляцию.

Arduino на столько быстро может подавать импульсы 5В в порт (1000 раз/сек), что будет казаться, что светодиод ярко светится, а если будет больше импульсов низкого уровня (LOW), то будет казаться что светодиод светится очень тускло. Так для формирования напряжения 2,5 вольта, количество логических нулей и единиц, за единицу времени, должно быть примерно равно. Баланс между HIGH и LOW определяет что мы "увидим" на выходе.





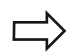
Код Arduino:



Откройте Arduino IDE // Файл > Примеры > MaxKit > lesson_3

На заметку:

```
for (x = 0; x < 768; x++)
{}
```



Цикл **for()** начинается с целого “x” равного 0, проверяет, остается ли “x” меньше 767, и, если так, увеличивает “x” на 1 (“x++”) и выполняет блок в фигурных скобках {}. Цикл повторяется, пока “x” не станет =768, затем выход из for()

```
if (x <= 255)
{}
else
{}
```

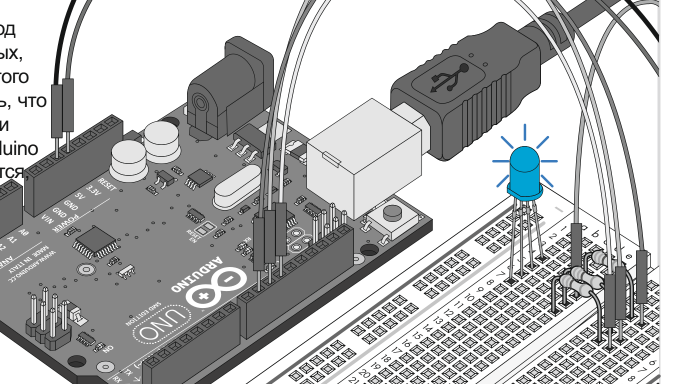
Конструкция **if**...**else** позволяет сделать выбор «если, тогда...». Оператор в скобках () оценивает условие: если условие ИСТИНА, "x<=255, выполняется код в первых скобках {}, а если НЕ ИСТИНА, "x=256", будет выполнен код во вторых скобках {}

```
delay(sensorValue);
```

Arduino работает очень, очень быстро, способна обрабатывать тысячи строк каждую секунду. Чтобы мы могли увидеть что произошло в программе, мы будем часто вставлять **Delay()**, код задержки. Рассчитывается в миллисекундах, 1000мс = 1сек.

Что вы должны увидеть:

Вы увидите, что ваш светодиод светится, но на сей раз в новых, сумасшедших цветах! Если этого не произошло, удостоверьтесь, что Вы собрали схемуправильно и проверили загружаемый в Arduino код. Если светодиод не светится, воспользуйтесь подсказками по поиску и устранению неисправностей.



Возможные трудности:

Светодиод не горит, или горит не тот цвет
Ножки светодиода находятся друг от друга на маленьком расстоянии, поэтому легко ошибиться, нужна повторная проверка подключения.

Сведения о красном
Красный светодиод может светиться более интенсивно, поэтому при необходимости воспользуйтесь дополнительным резистором, или измените код:

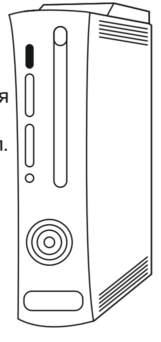
analogWrite(RED_PIN, redIntensity);

на

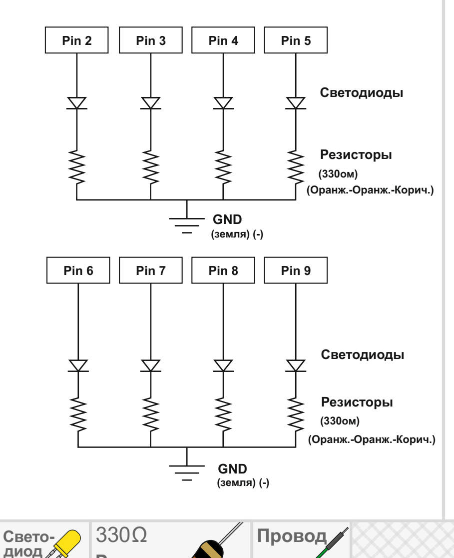
analogWrite(RED_PIN, redIntensity/3);

Применение в жизни:

Применяется в световых шоу, а также в различных видах электроники, для того чтобы в той же самой области показать изменения состояния устройства. Например из состояния Вкл. на Выкл.



ОПЫТ #4



4

Множество Светодиодов

Фантастика - Вы заставили мигать светодиод! Теперь пора усложнить схему и подсоединить к Arduino сразу ВОСЕМЬ СВЕТОДИОДОВ. Таким образом, с помощью Arduino, вы создадите различные световые эффекты. Схема достаточно большая и интересная. В этом опыте вы начнете практиковать запись собственных программ и получите представление о том, как работает Arduino.

Наряду с управлением СЕТОДИОДАМИ, Вы научитесь приемам программирования, которые сделают Ваш код понятным и аккуратным:

for() - (цикл) - используется, когда Вы хотите выполнить часть кода несколько раз.

arrays[ ] - (массив) - упрощает код, и делает доступ к переменным проще, собирая их в группы.

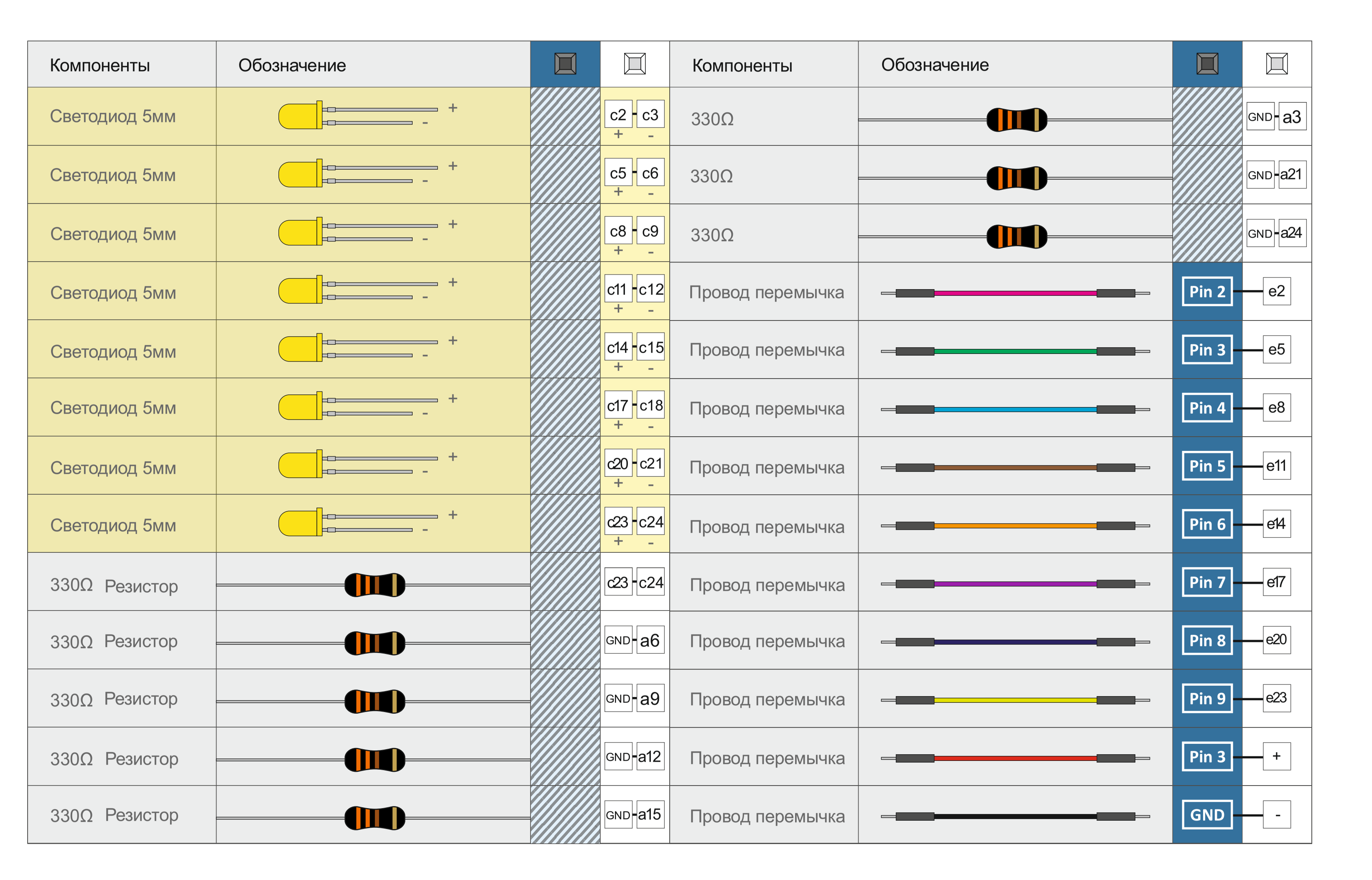
ДЕТАЛИ:

Свето-диод x8

330Ω Резистор x8

Провод x10

Опыт 4: Множество светодиодов



Компоненты	Обозначение			Компоненты	Обозначение		
Светодиод 5мм			c2 (+) - c3 (-)	330Ω			GND - a3
Светодиод 5мм			c5 (+) - c6 (-)	330Ω			GND - a21
Светодиод 5мм			c8 (+) - c9 (-)	330Ω			GND - a24
Светодиод 5мм			c11 (+) - c12 (-)	Провод перемычка		Pin 2	e2
Светодиод 5мм			c14 (+) - c15 (-)	Провод перемычка		Pin 3	e5
Светодиод 5мм			c17 (+) - c18 (-)	Провод перемычка		Pin 4	e8
Светодиод 5мм			c20 (+) - c21 (-)	Провод перемычка		Pin 5	e11
Светодиод 5мм			c23 (+) - c24 (-)	Провод перемычка		Pin 6	e14
330Ω Резистор			c23 - c24	Провод перемычка		Pin 7	e17
330Ω Резистор			GND - a6	Провод перемычка		Pin 8	e20
330Ω Резистор			GND - a9	Провод перемычка		Pin 9	e23
330Ω Резистор			GND - a12	Провод перемычка		Pin 3	+
330Ω Резистор			GND - a15	Провод перемычка		GND	-





Откройте Arduino IDE // Файл > Примеры > MaxKit > lesson_4

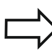
На заметку:

`int ledPins[] = {2,3,4,5,6,7,8,9};`



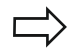
“Массив” - самый удобный способ управлять множеством переменных, для этого их достаточно объединить в группы. Здесь мы создаем массив из целых чисел, с названием ledPins, состоящий из восьми элементов.

`digitalWrite(ledPins[0], HIGH);`



Вы обращаетесь к элементам в массиве по их положению. Первый элемент в массиве находится в позиции 0, второй в 1-ой, третий во 2-ой и т. д. Обратиться к элементу можно с помощью "ledPins [x]", где [x] это позиция. Здесь мы подаем +5 В. на Pin2, потому что в позиции “0” у нас порт 2.

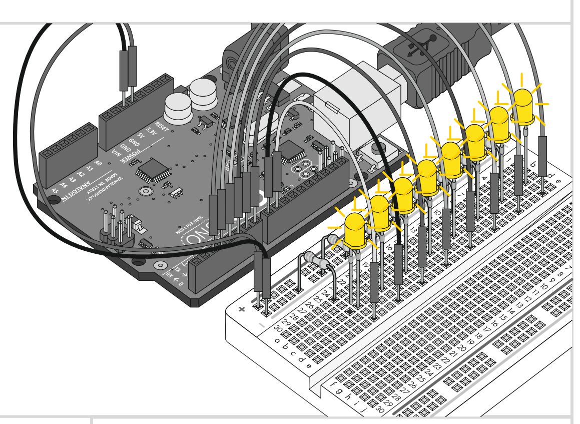
`index = random(8);`



Компьютеры любят делать одно и тоже много, много раз. Но иногда вы хотите прервать цепь однообразия и вызвать элемент случайным образом, например, имитируя игру в кости. Для этого существует Функция random().

Что вы должны увидеть:

Этот пример похож на первый опыт, но вместо одного светодиода, вы должны увидеть множество мигающих светодиодов. Если этого не произошло, убедитесь в том, что вы правильно собрали схему и проверьте загружен ли код в Arduino, или смотрите советы по устранению неполадок ниже.



Возможные трудности:

Некоторые светодиоды не загораются.
Распространенной ошибкой является неправильная полярность светодиода, извлеките его, поверните на 180 градусов, и снова вставьте

Нарушена последовательность.
В восьми проводах очень легко ошибиться. Дважды проверьте, что первый светодиод подключен к порту 2, второй к 3-му и т. д.

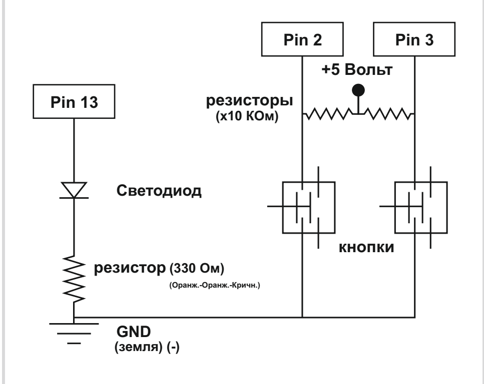
Начните все заново.
Иногда начать собирать схему заново намного легче, чем отыскать ошибку в хитросплетении огромного количества проводов.

Применение в жизни:

Бегущая строка, - все вы конечно видели информационные табло на вокзале, или в аэропорту, как раз они состоят из множества светодиодов.

5

Кнопки

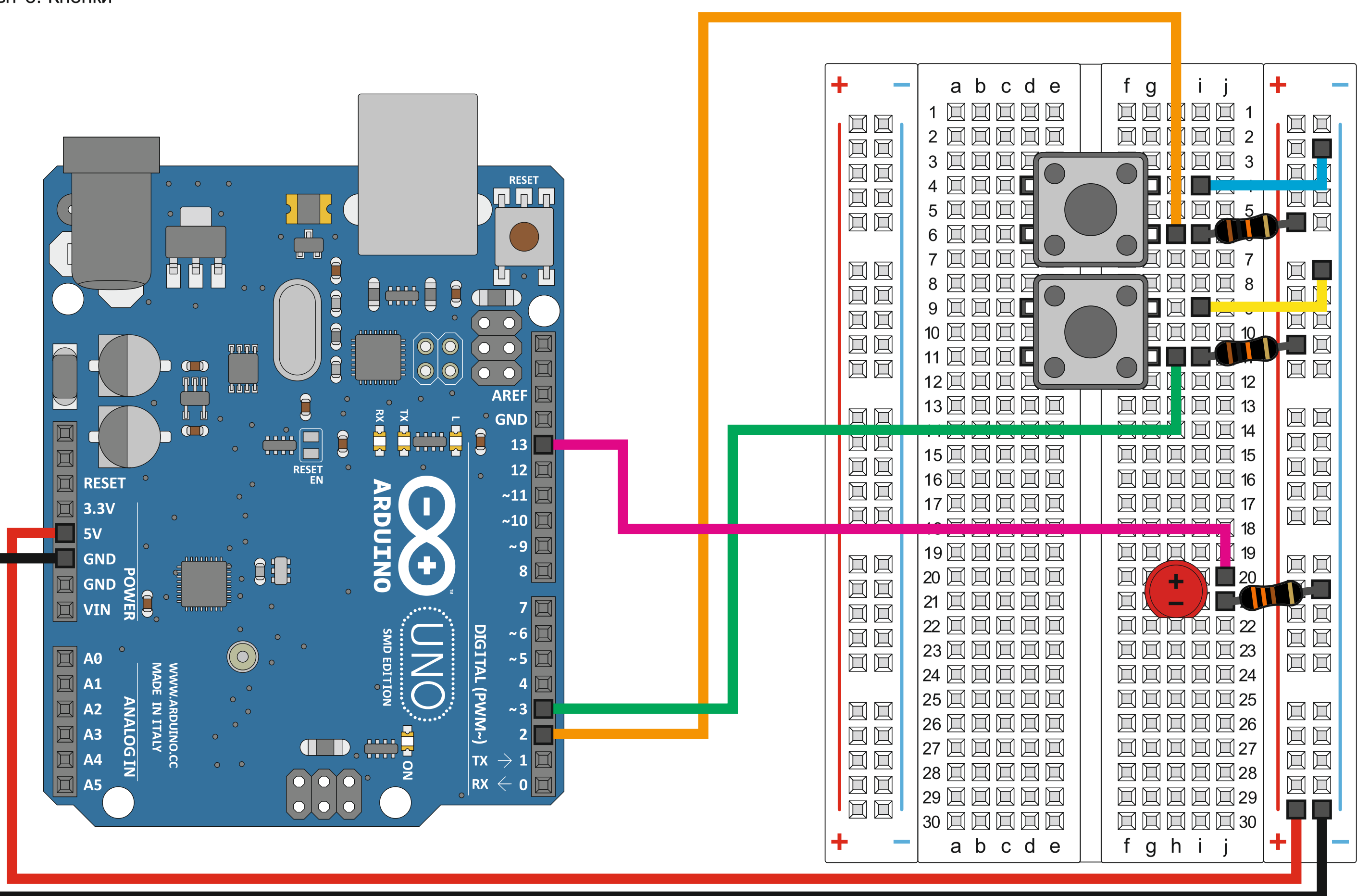


До сих пор мы занимались исключительно выходами. Теперь мы собираемся зайти с другой стороны, и поиграем с входами. В этой схеме, в качестве датчика, мы будем использовать кнопки - наиболее простой способ получения входной информации. В данном случае кнопки подключены между "Землей" и Arduino, и если кнопка нажата, вход Arduino замыкается на "0" (LOW). Arduino считывает это и реагирует соответствующим образом.
В этой схеме вы также можете заметить "подтягивающие" резисторы, которые помогают "очистить" сигнал и предотвратить ложные срабатывания кнопки.

детали:

Деталь	Количество
Кнопка	x2
Свето-диод	x1
10KΩ Резистор	x2
330Ω Резистор	x1
Провод	x7

Опыт 5: Кнопки



Компоненты	Обозначение	Arduino	Макетная плата
Кнопка			d4 g4 d6 g6
Кнопка			d9 g9 d11 g11
Светодиод	+ -		h20 (+) h21 (-)
10K Резистор			i6 +
10K Резистор			i11 +
330Ω Резистор			j21 -
Провод перемычка			i4 -
Провод перемычка			i9 -
Провод перемычка		Pin 2	h6
Провод перемычка		Pin 3	h11
Провод перемычка		Pin 13	j20

Использование логики.

Одна из вещей, которая делает Arduino чрезвычайно полезной, это то, что она может решить сразу комплекс проблем. Например, из нее можно сделать термостат, который включает обогрев, если становится слишком холодно, вентилятор, если становится слишком жарко, полить растения, если почва становится слишком сухой и т.д.

Для того, чтобы принимать важные решения, у Arduino есть набор логических операций, которые в зависимости от поставленных задач позволяют решить, что же делать дальше, "если" выполняется заданное условие.

==	**ЭКВИВАЛЕНТНОСТЬ**	**A == B** - true (истина), если A и B **равны**
!=	**НЕ РАВНО**	**A != B** - true (истина), если A и B **не равны**
&&	логическое **“И”**	**A && B** - истина, если оба и **A и Б - истина**
||	логическое **“ИЛИ”**	**A || B** - истина, если один из **A или B истина**
!	**ОТРИЦАНИЕ**	**A ! B** - превращает ложь в истину, и наоборот

Вы можете комбинировать эти функции для того, чтобы создавать дополнительные условия if().

Пример:

```
if ((режим == обогрев) && ((температура < уст.значение) || (ручн.коррекц. == истина)))
{
digitalWrite(ОБОГРЕВАТЕЛЬ, ВКЛЮЧИТЬ);
}
```

...включить нагреватель, если установлен режим обогрева, “И” температура понизилась, “ИЛИ” если вы включили обогрев в ручном режими. Используя эти логические операторы, вы можете запрограммировать ваш Arduino для принятия интеллектуальных решений и взять под контроль мир вокруг Вас!



Код Arduino:

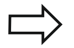


Откройте Arduino IDE // Файл > Примеры > MaxKit > lesson_5

На заметку:

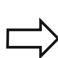


`pinMode(button2Pin, INPUT);`



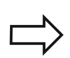
Цифровые порты, или по другому - пины могут быть как входящими, так и исходящими, правда сначала об этом нужно сообщить Arduino

`button1State = digitalRead(button1Pin);`



Для считывания данных с цифровых портов используется функция digitalRead(), она вернет значение HIGH, если присутствует 5 Вольт, и значение LOW, если присутствует 0 Вольт.

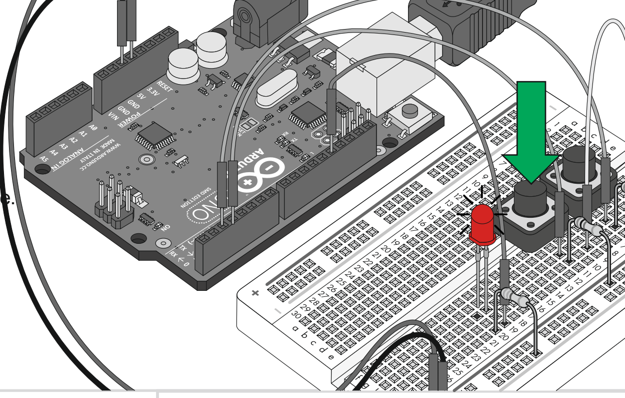
`if (button1State == LOW)`



Когда кнопка нажата, одним контактом она замыкается на “Землю”, это распознается Arduino как “LOW”. Здесь мы используем оператор сравнения в условии “if”, ("=="), который сравнивает нажата кнопка или нет.

Что вы должны увидеть:

Вы должны увидеть горящий светодиод, когда нажата кнопка. Если этого не происходит, убедитесь что схема собрана правильно, а также загружен ли код в Arduino, в противном случае обратитесь к разделу устранения неисправностей ниже.



Возможные трудности:

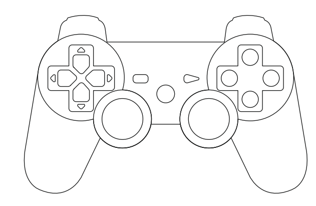
Свет не включается
Кнопки квадратные и легко ошибиться при подключении их в плату, попробуйте повернуть их на 90 градусов.

Свет не выключается
Простые ошибки может совершить каждый, посмотрите не подключен ли светодиод к порту 9, вместо 13.

Восторг
Не беспокойтесь, все элементы этих схем чрезвычайно просты и надежны, их можно повредить, разве что разбив.

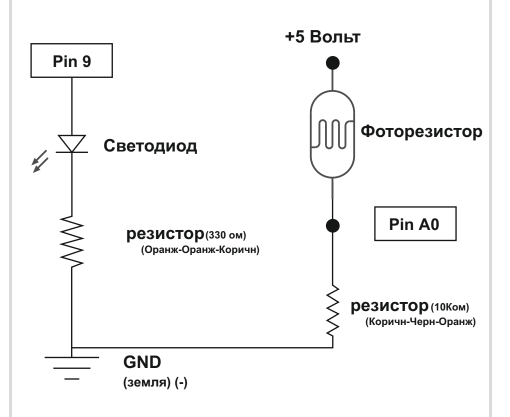
Применение в жизни:

Все вы наверное хорошо представляете где могут использоваться кнопки - пульты, джойстики, телефоны и т. д..



ОПЫТ 6

Фоторезистор



Итак, вы уже позанимались с потенциометром, сопротивление которого изменяется из за поворота рукоятки - штока. В этой схеме, вы познакомитесь с фоторезистором, который изменяет свое сопротивление в зависимости от того, сколько света на него направлено. Arduino не может сама интерпретировать сопротивление, т.к работает с напряжением, поэтому в этой схеме мы будем использовать делитель напряжения. Делитель состоит из двух резисторов, один из которых будет наш фоторезистор, а считываемое напряжение берется из точки между ними. Делитель будет выдавать высокое напряжение- когда фоторезистор получает много света, и низкое- когда фоторезистор получает мало света.

детали:

Фоторезистор	Светодиод	330 Ом Резистор	Провод	10K Резистор
x1	x1	x1	x6	x1

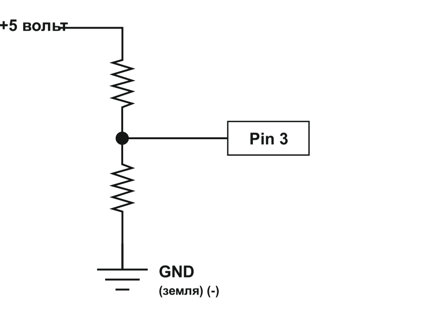
Опыт 6 : Фоторезистор

Компоненты	Обозначение	Arduino	Макетная плата
Фоторезистор			f5 - f6
Светодиод ⚠	+ -		h20 (+) - h21 (-)
330 Резистор			i21 - (-)
10K Резистор			i1 - i5
Провод перемычка			j1 - (+)
Провод перемычка		A0	j5
Провод перемычка			j6 - (-)
Провод перемычка		Pin 9	j20
Провод перемычка		5V	+
Провод перемычка		GND	-

Снятие показаний с резистивных датчиков:

Многие из датчиков, которые вы будете использовать (фоторезисторы, потенциометры и пр.), есть не что иное, как резистор в той или иной степени. Их сопротивление изменяется в зависимости от чего либо (у фото-резистора от уровня освещенности, у потенциометра от поворота его ручки и т. д.).

Аналоговые входы Arduino работают с напряжением, а не сопротивлением. Мы можем это легко исправить, включив резистивный датчик как часть "делителя напряжения".



Делитель напряжения состоит из двух резисторов."Сверху" резистор ДАТЧИК, который Вы будете использовать."Снизу" один обычный резистор, с фиксированным номиналом. Когда вы подключите верхний резистор к +5 вольт, а нижний резистор к "земле", напряжение в средней точке будет меняться пропорционально номиналу этих резисторов. Т.е при изменении сопротивления фото-резистора, в зависимости от освещенности, будет меняться выходное напряжение в средней точке делителя, что нам и надо!
В качестве резистора для нижнего плеча, мы используем резистор 10 КОм. Во всяком случае такой номинал наиболее универсален для большинства наших экспериментов.



Код Arduino:



Откройте в Arduino IDE // Файл > Примеры > MaxKit.ru > *lesson_6*



На заметку:

```
lightLevel = map(lightLevel, 0, 1023, 0, 255);
```

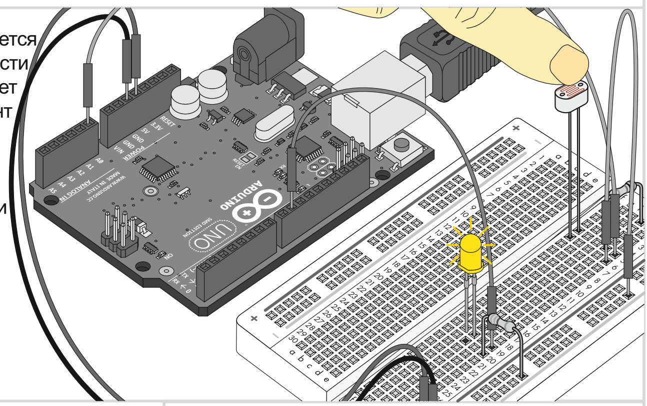
Когда мы считываем аналоговый сигнал, используя analogRead(), диапазон получается от 0 (0 вольт) и до 1023 (5 вольт). Мы же хотим управлять функцией analogWrite(), а ее рабочий диапазон от 0 до 255. Преобразовать один больший диапазон, в меньший поможет функция map().

```
lightLevel = constrain(lightLevel, 0, 255);
```

Поскольку map() может вернуть числа, находящиеся за пределами диапазона, скажем: или больше, или меньше, нам понадобится функция constrain(). Она проверит, содержатся ли числа в заданном диапазоне. Если число находится вне диапазона, она уменьшит или увеличит его до заданного, а если в пределах, то оно останется неизменным.

Что вы должны увидеть:

Вы должны увидеть,как изменяется яркость светодиода в зависимости от того, как много света попадает на фоточувствительный элемент вашего фоторезистора.
Если этого не происходит, убедитесь, что схема собрана правильно. А также загружен ли код программы в Arduino.
Если и это не помогло, воспользуйтесь советами по устранению неполадок ниже.



Возможные трудности:

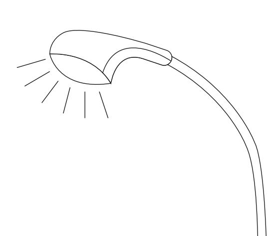
Светодиод не светится
Это ошибка, которую мы будем делать снова и снова, пока не появятся светодиоды, которые не имеют полярности (+/-). Поверните светодиод.

Схема не реагирует на освещение
Учитывая что у фоторезистора связь со схемой производится проводами, в них очень легко ошибиться. Проверьте правильность соединений.

Все еще не работает
Вы можете находиться в комнате с сильным или наоборот с недостаточным освещением. Попробуйте воспользоваться фонариком.

Применение в жизни:

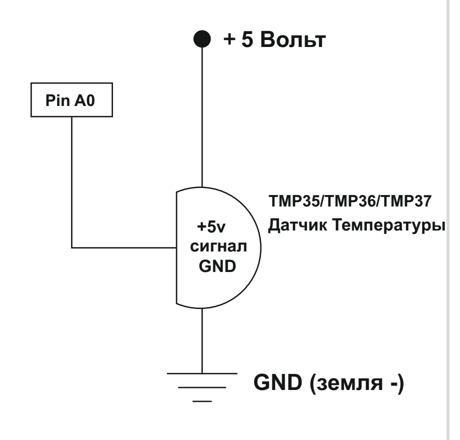
Уличные фонари зажигаются с наступлением темноты.



ОПЫТ 7

Датчик Температуры

7



Датчик температуры - правильнее будет назвать датчиком для измерения температуры окружающей среды. Данный датчик имеет три контакта - положительный, сигнал и земля. Он вырабатывает напряжение пропорционально температуре в градусах Цельсия (коэффициент преобразования ~10 мВ/°C). В этом опыте вы узнаете, как использовать датчик температуры с платой Arduino, а также как использовать окно отладки ArduinoIDE для отображения температуры.



Когда вы будете создавать эту схему, будьте осторожны, не перепутайте датчик температуры и транзистор, их корпуса очень похожи.

детали:

- Датчик TMP36 x1
- Провод x5

Опыт 6: Датчик Температуры

Компоненты	Обозначение	Arduino	Макетная плата
Датчик температуры			f5 f6 f7
Провод перемычка			j5 -
Провод перемычка		A0	j6
Провод перемычка			j7 +
Провод перемычка		5V	+
Провод перемычка		GND	-

Как открыть монитор последовательного порта:

Перед тем, как открыть монитор порта, вы должны загрузить код (1) программы в плату Arduino, и только затем открыть Монитор порта (2). Только после этого откроется дополнительное окно (3).





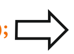
Откройте в Arduino IDE // Файл > Примеры > MaxKit.ru > lesson_7

На заметку:

Serial.begin(9600);  Для того, чтобы обмен данными происходил корректно, необходимо соблюсти общий протокол и скорость обмена данными. Передача данных измеряется в битах в секунду. Это происходит при инициализации функции Serial.begin(9600).

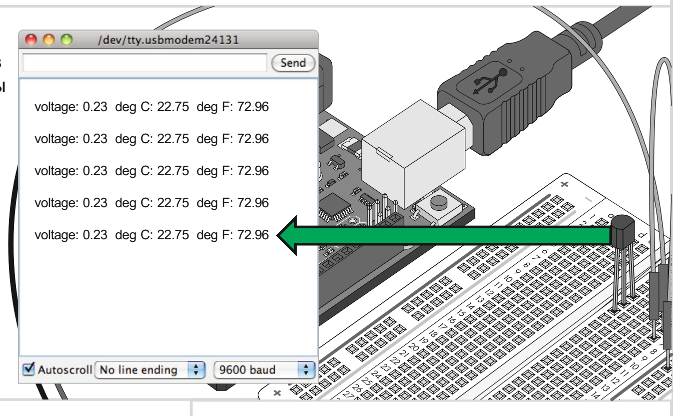
Serial.print(degreesC);  Serial.print() очень умная команда, которая может напечатать практически все, что вы ей дадите, в том числе переменные различных типов, а также ваш текст, например "strings", и пр.

Дополнительно смотрите http://arduino.cc/en/Serial/Print

Serial.println(degreesF);  Serial.print() печатает текст на одной сроке, а вот Serial.println() в конце переносит курсор на следующую строку. Обычно эти команды используют вместе, чтобы иметь возможность получить удобно читаемые куски текста и данных.

Что вы должны увидеть:

В результате проведенного опыта, вы должны увидеть в окне “монитора порта” среды программирования Arduino IDE, надписи подобно тем, которые вы видите на этом рисунке.
Если этого не происходит, проверьте правильность собранной вами схемы.



Возможные трудности:

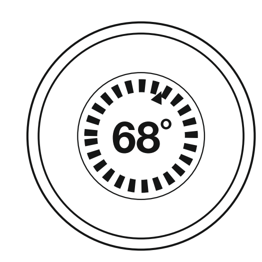
Кажется ничего не происходит
Выполняемая программа не имеет внешних признаков выполнения. Чтобы увидеть результат, откройте “Монитор порта” среды Arduino IDE.

На дисплее какая-то “тарабарщина”.
Такое может происходить из за не соблюдения скорости передачи данных между Arduino и последовательным портом. Установите скорость в окне “Монитора порта” - “9600 baud”.

Значение температуры не меняется
Попробуйте прикоснуться к датчику пальцем, тем самым вы немного нагреете его.

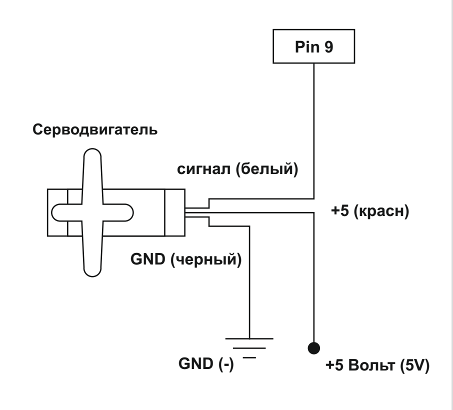
Применение в жизни:

В системах кондиционирования воздуха - датчик передает показания на пульт управления.



ОПЫТ 8

8



Серводвигатель

"Серво", сокращение от серводвигателя, это двигатель, который включает в себя электронную схему с обратной связью. Серводвигатели идеально подходят для применения в электронике и автоматике, потому что обладают высокими скоростными характеристиками и умеют очень точно устанавливаться в нужное положение, что не умеют обычные двигатели.
Достаточно простое управление, при помощи импульсов переменной длительности. Например, импульс 1,5 мс установит сервопривод в положение 90 градусов.

В этом опыте вы узнаете, как использовать ШИМ (широтно-импульсную модуляцию) для управления серводвигателем.

детали:

Серво x1

Провод x8

Опыт 8: Серводвигатель

Компоненты	Обозначение		
Серводвигатель			e5 e6 e7
Провод перемычка			e5
Провод перемычка			e6
Провод перемычка			e7
Провод перемычка		Pin 9	a7
Провод перемычка			b5 -
Провод перемычка			a6 +
		5V	+
		GND	-

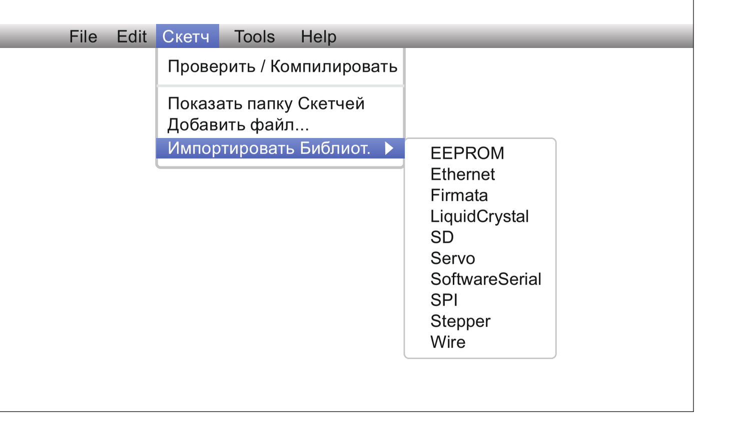
Расширьте свои возможности с помощью библиотек:

У Arduino имеется полезный набор встроенных команд для выполнения сновных операций ввода / вывода. Она может принимать решения, используя логику, а также решать математические задачи и пр.. Но реальной силой Arduino является огромное сообщество заинтересованных людей, а также их готовность поделиться результатами своей работы.

Библиотеки- это сборники новых команд, которые собраны вместе для того, чтобы их можно было легко подключить к вашему проекту. Ардуино поставляется с несколькими полезными библиотеками, такими как Wire, LiquidCrystal, Ethernet, Servo и пр.. Servo - библиотека сервоприводов, как раз используемая в этом опыте. Дополнительную информацию см. на:

http: //arduino.cc/en/Reference/Libraries список библиотек и информации об их использовании.

Если вы собираетесь использовать новый датчик или устройство, есть вероятность того, что кто-то уже создал библиотеку, для взаимодействия с вашим устройством. Для многих датчиков и устройств библиотеки давно созданы, их остается только найти. Вам в помощь Google и Yandex. Чтобы использовать библиотеку в проекте, выберите его из меню ArduinoIDE > **Sketch > Import Library (Скетч> Импортировать Библиотеку)**





Код Arduino:



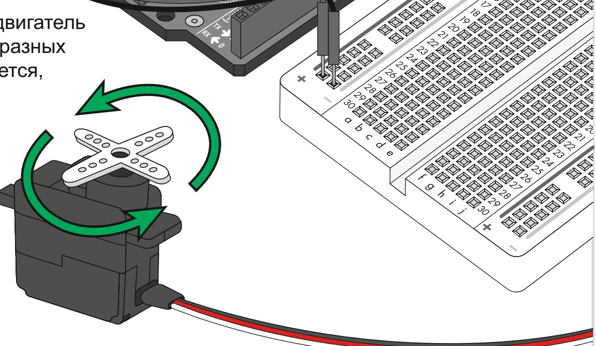
Откройте в Arduino IDE // Файл > Примеры > MaxKit.ru > lesson_7

На заметку:

#include <Servo.h>		#include специальная директива препроцессора, которая вставляет библиотеку (или любой другой файл) в ваш проект. Вы можете ввести эту команду вручную, или выбрать установленную библиотеку из меню ArduinoIDE: Скетч> Импортировать Библиотеку> Add Library.
Servo servo1; servo1.attach(9);	⇨	Библиотека Servo содержит новые команды, которые позволяют управлять различными сервоприводами. Чтобы подготовить Arduino для управления серводвигателем, вы должны сначала создать «объект» Servo для каждого сервопривода (мы назвали его "servo1"), а затем прикрепить (attach) его к цифровому порту (у нас это порт 9).
servo1.write(180);	⇨	Сервоприводы не умеют вращаться по кругу, но умеют занимать определенную позицию. Мы используем команду write() библиотеки Servo, чтобы установить сервопривод в указанное число градусов (от 0 до 180). Помните, что сервоприводу требуется время, чтобы достичь определенной позиции, поэтому дайте ему задержку delay().

Что вы должны увидеть:

Вы должны увидеть, как ваш серводвигатель переходит в различные позиции на разных скоростях. Если двигатель не двигается, проверьте соединения и убедитесь, что код загружен в Arduino, или воспользуйтесь советами по устранению неполадок ниже.



Возможные трудности:

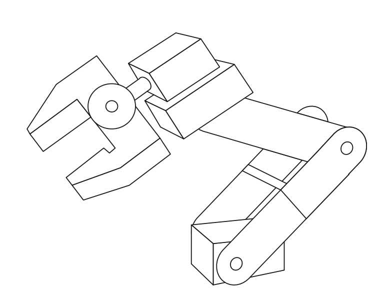
Servo не крутится
Даже с цветными проводами есть шанс подключить серво в обратном направлении. Проверьте правильность соединений.

Тем не менее не работает
Ошибка, которую мы сделали несколько раз - забыли подключить провода питания +5 вольт и землю.

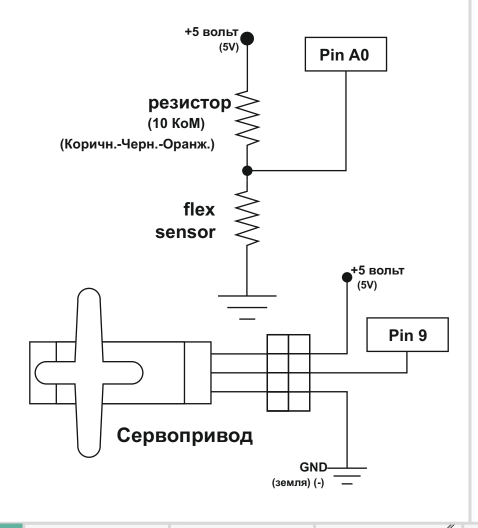
Двигается рывками
Если серво начинает двигаться рывками и мигает светодиод на борту Arduino, это говорит о том, что не хватает мощности питания. Воспользуйтесь сетевым адаптером вместо USB, это должно решить проблему.

Применение в жизни:

Роботизированные механические руки, которые используются на автомобильных заводах.



ОПЫТ 9



9

Датчик изгиба

В этом опыте мы используем Flex Sensor, или "Датчик Изгиба". Его основа - пластиковая полоска с нанесенным слоем угля. Также как у переменного резистора, у него меняется сопротивление, но вместо поворота ручки, вам приходится сгибать его, тем самым изменяя сопротивление. Для обнаружения этих изменений, мы снова используем "делитель напряжения". Наклоняя "Датчик" в одном направлении, он изгибается, и чем больше он изгибается, тем выше становится его сопротивление. Приблизительный интервал изменений от 10 кОм до 35 ком. В этой схеме мы будем использовать Flax Sensor для контроля положения сервопривода.

ДЕТАЛИ:

Flex Sensor	Серво	Резистор 10 Ком	Провод
x1	x1	x1	x11

+ **Если в вашем наборе отсутствует датчик изгиба, вы можете заменить его, не меняя схемы, фоторезистором. Подправив код. см. коментарий в скетче.**

Опыт 9: Flex Sensor (Датчик Изгиба)

Компоненты	Обозначение	Arduino	Макетная плата
Серводвигатель			e1 e2 e3
Провод перемычка			e1 - ●
Провод перемычка			e2 - ●
Провод перемычка			e3 - ○
Flex Sensor			h19 - h20
10K Резистор			i20 - i24
Провод перемычка			i19 - -
Провод перемычка		A0	j20
Провод перемычка			h24 - +
Провод перемычка			b1 - -
Провод перемычка			a2 - +
Провод перемычка		Pin 9	a3
Провод перемычка		5V	+
Провод перемычка		GND	-

Отладка скетча с использованием Монитора порта:

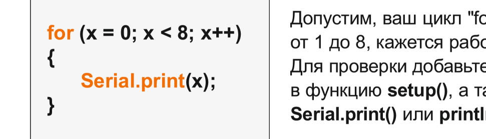
Такое может случиться с каждым: программа написана, успешно скомпилирована, загружена, но не понятно почему, работает не так как задумано. У компьютера есть монитор, клавиатура, мышь, с их помощью легко и удобно отлаживать код, но у Arduino подобных аксессуаров нет.

“Ключ" - в видимости состояния выходов микроконтроллера. Этим ключиком может стать практически что угодно - светодиод, зуммер, и пр., но один из самых полезных инструментов Serial Monitor. Используя Serial.print() и println(), вы можете с легкостью вывести, в понятном для человека виде, текст и данные от Arduino, на экран компьютера. Это отличная возможность для проверки и окончательной отладки Вашей программы.

```
for (x = 0; x < 8; x++)
{
    Serial.print(x);
}
```

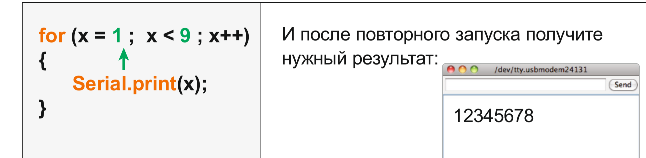
Допустим, ваш цикл "for()" со счетчиком от 1 до 8, кажется работает не правильно. Для проверки добавьте **Serial.begin(9600);** в функцию **setup()**, а также добавьте **Serial.print()** или **println()** в **loop()**.

Вы хотели от 1 до 8, но счетчик, на самом деле ситает от 0 до 7. Следовательно нужно подправить код.




```
for (x = 1 ;  x < 9 ; x++)
{
    Serial.print(x);
}
```

И после повторного запуска получите нужный результат:





Код Arduino:

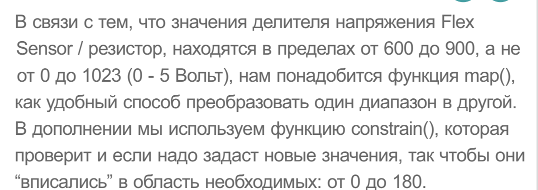


Откройте в Arduino IDE // Файл > Примеры > MaxKit.ru > lesson_9

На заметку:

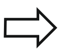
```
servoposition = map(flexposition, 600, 900, 0, 180);
servoposition = constrain(servoposition, 0, 180);
```





В связи с тем, что значения делителя напряжения Flex Sensor / резистор, находятся в пределах от 600 до 900, а не от 0 до 1023 (0 - 5 Вольт), нам понадобится функция map(), как удобный способ преобразовать один диапазон в другой. В дополнении мы используем функцию constrain(), которая проверит и если надо задаст новые значения, так чтобы они “вписались” в область необходимых: от 0 до 180.

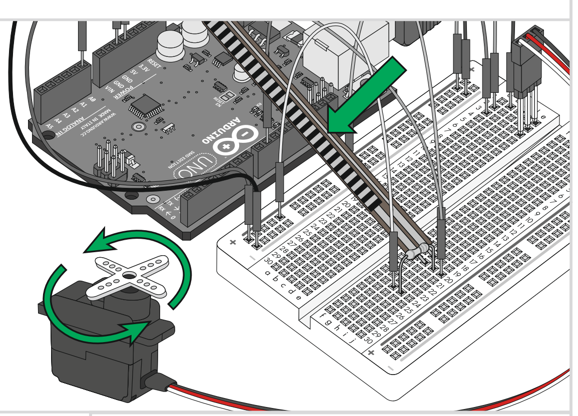
```
Serial.print("sensor: ");
Serial.print(flexposition);
Serial.print("  servo: ");
Serial.println(servoposition);
```



Serial.print(”sensor: ”); напечатает **sensor:**, добавит значение из переменной flexposition, далее Serial.print(” servo: ”); напечатает **servo:**, с пробелом перед словом servo, добавит значение из переменной servoposition, и после этого, из за Serial.print**ln**, переведет курсор на новую строку.

Что вы должны увидеть:

Вы должны увидеть поворот серводвигателя в соответствии с тем, на сколько много вы изогнули гибкий датчик. Если схема не работает, убедитесь в том, что все собрано правильно, а также загружен ли код в Arduino, или просмотрите советы по устранению неполадок ниже.



Возможные трудности:

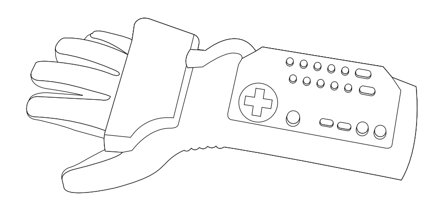
Servo не поворачивается
Даже в цветных проводах идущих от серво легко ошибиться. Возможно ошибка в этом.

Servo не движется, как ожидалось
Датчик предназначен для изгиба в одном направлении, попытайтесь изгибать полоску в противоположную сторону.

Малый диапазон поворота Servo
Вам нужно попытаться изменить диапазон значений в функции map().

Применение в жизни:

Яркий пример применения датчиков изгиба- это игровые консоли Nintendo “Power Glove”. Это был первый игровой видео контроллер, который использовал имитацию движения рук на экране в реальном времени.

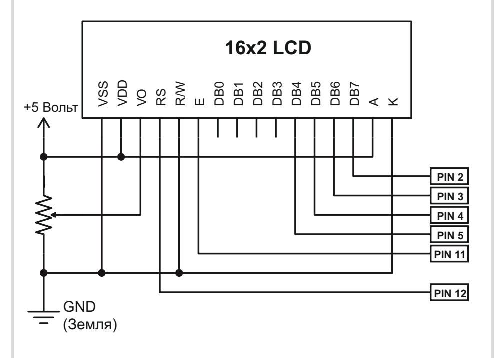


ОПЫТ 10

10

ЖКИ (LCD)

Жидко Кристаллический Индикатор



В этом опыте Вам предстоит освоить ЖКИ (LCD*). ЖКИ, или жидкокристаллический дисплей является простым дисплеем, который может отображать выполнение команд, выводить информацию от различных датчиков - все зависит от того, как Вы запрограммируете свою плату Arduino.
Вы узнаете основные принципы работы с ЖКИ: как подключить, как настроить, как использовать ЖКИ в Ваших проектах с Arduino.
ЖКИ модуль поставляется отдельно от разъема и если у вас возникнут сложности с его пайкой, - обратитесь за помощью к знающему человеку, или специалисту.

*LCD Liquid Crystal Display (eng.)

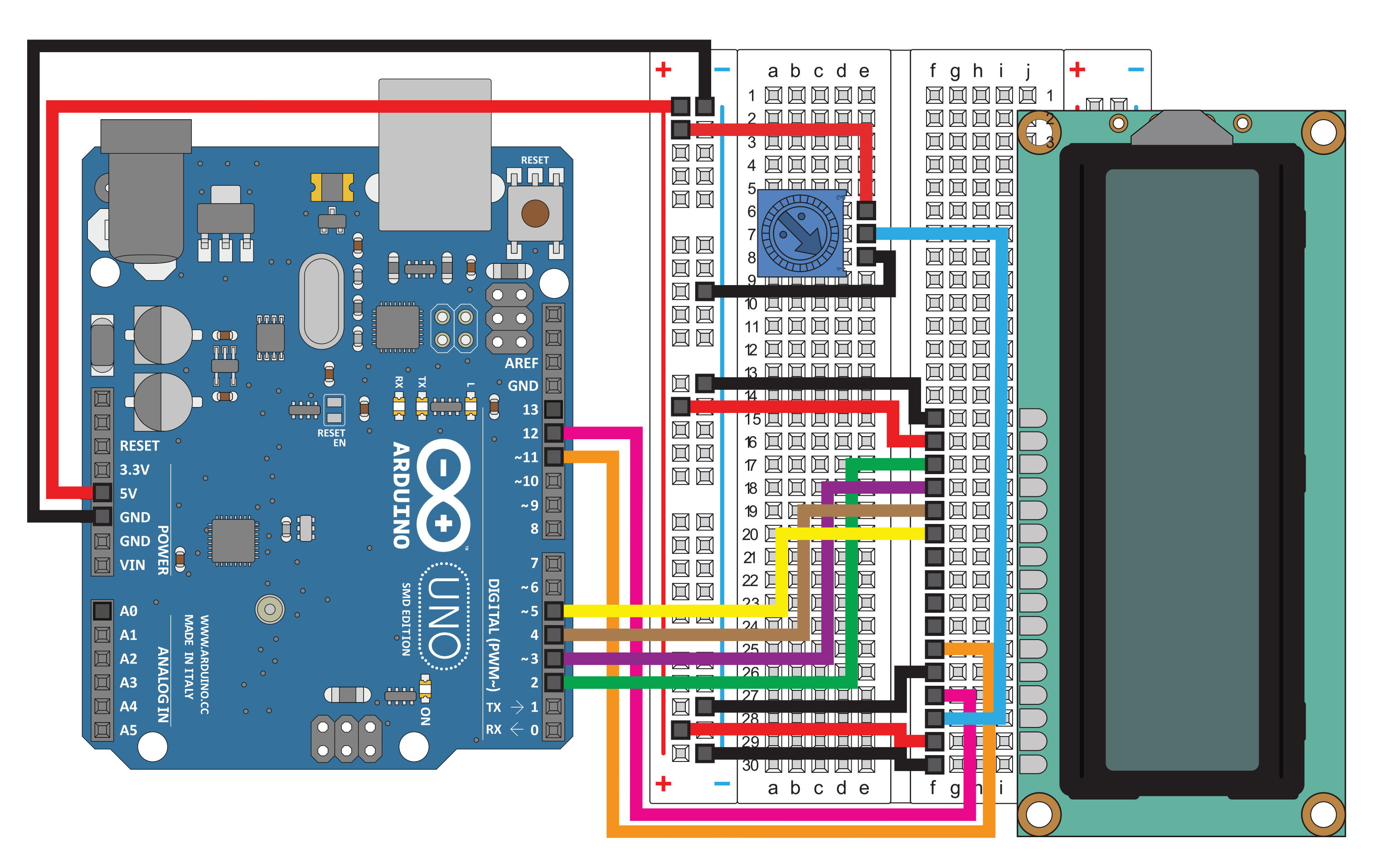
ДЕТАЛИ:

Потенциометр x1

ЖКИ x1

Провод x16

Опыт 10: ЖКИ



RESET
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX 1
RX 0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
SMD EDITION
ON
RESET EN
WWW.ARDUINO.CC
MADE IN ITALY
POWER
RESET
3.3V
5V
GND
GND
VIN
ANALOG IN
A0
A1
A2
A3
A4
A5
a b c d e
f g h i j

Компоненты	Обозначение		
ЖКИ			j30, j29, j28, j27, j26, j25, j24, j23, j22, j21, j20, j19, j18, j17, j16, j15
Потенциометр			b8, b7, b6
Провод перемычка		+5V	+
Провод перемычка		GND	-
Провод перемычка			e6 - +
Провод перемычка			e7 - f28
Провод перемычка			e8 - -
Провод перемычка			f15 - -
Провод перемычка			f16 - +
Провод перемычка		Pin 2	f16

Компоненты	Обозначение		
Провод перемычка		Pin 3	f 18
Провод перемычка		Pin 4	f 19
Провод перемычка		Pin 5	f 20
Провод перемычка		Pin 11	f 25
Провод перемычка			f 26 - -
Провод перемычка		Pin 12	f 27
Провод перемычка			f 29 - -
Провод перемычка			f 30 - -



Код Arduino:



Откройте в Arduino IDE // Файл > Примеры > MaxKit.ru > lesson_10

На заметку:

```
#include <LiquidCrystal.h>
```



Этот кусок кода говорит Arduino IDE о том, что перед тем как начать работу с программой, необходимо подключить библиотеку LiquidCrystal.h. Она упрощает управление различными типами жидкокристаллических индикаторов.
Без нее никакие команды работать не будут! Убедитесь, что библиотека подключена.

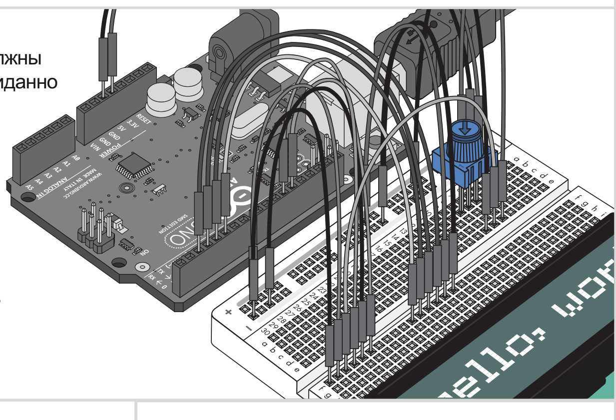
```
lcd.print("hello, world!");
```



В первый раз, после включения, вместо каких либо символов, Вы можете увидеть на дисплее яркое свечение и больше ничего. Не отчаивайтесь, всего-навсего Вам нужно отрегулировать яркость ЖК индикатора. Для этого покрутите ручку потенциометра и добейтесь появления на экране надписи “helo, world!”.

Что вы должны увидеть:

Первоначально на дисплее вы должны увидеть слова "hello, world!", неожиданно появившиеся на вашем дисплее. Помните?, вы можете настроить контрастность с помощью ручки потенциометра, если слова на дисплее не очень разборчивы. Если возникли сложности, убедитесь, что код программы написан правильно, и дважды проверьте правильность всех соединений.



Возможные трудности:

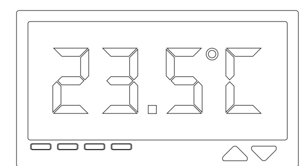
Экран ярко светится
Попытайтесь отрегулировать яркость ручкой потенциометра, в противном случае вы не сможете прочесть текст на экране ЖК.

Вообще не работает
Тщательно проверьте правильность написания кода программы, в частности подключение библиотеки LiquidCrystal.

Мерцание экрана
Тщательно проверьте правильность соединения вашей платы и Arduino, а также цепей питания.

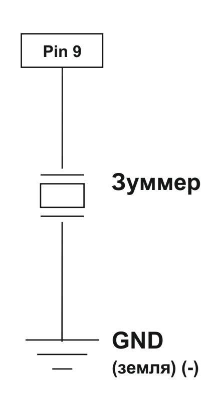
Применение в жизни:

ЖК дисплеи сегодня есть везде! Часы на вашей руке, табло в микроволновке, даже современный телевизор- есть не что иное, как Жидкокристалический Дисплей!



ОПЫТ 11





ЗУММЕР, ПИЩАЛКА (англ. Buzzer)

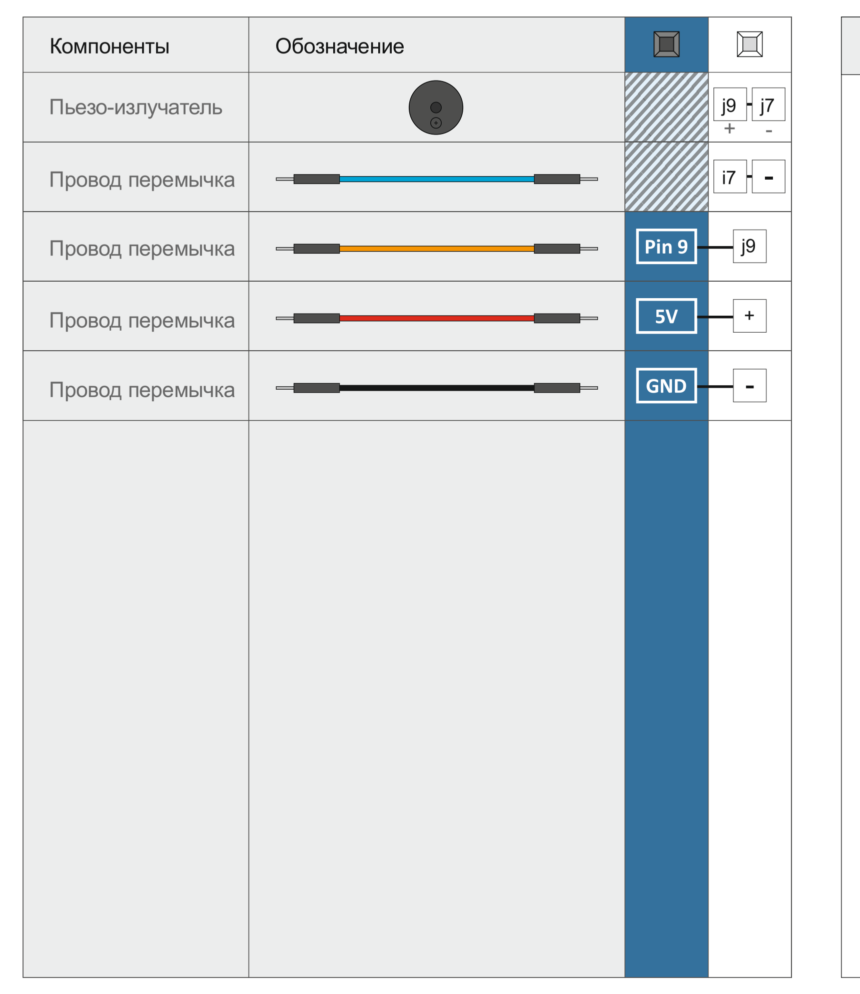
В этом опыте, мы снова будем преодолевать разрыв между миром цифровым и аналоговым. Мы будем использовать зуммер (пищалку), который делает небольшой «щелчок», если дотронутся, кратковременно, его контактами до питания +5 вольт и до “земли”, попробуйте! Само по себе это не очень интересно, но если вы подадите на него напряжение и тут же отключите, со скоростью 100 раз в секунду, зуммер начнет пищать. И если собрать сотни строк тонов вместе, у вас появится музыка!

В этом опыте схема, вместе со скетчем, будет играть классическую музыку. Во всяком случае мы на это надеемся!

ДЕТАЛИ: Пьезо-элемент x1 | Провод x4

Если зуммер не вписывается в отверстия на плате, попробуйте его немного повернуть, так чтобы выводы вошли в отверстия по диагонали.

Опыт 11: Пищалка



Создание ваших собственных функций:

Arduino содержит множество полезных функций, которые существуют для чего угодно. (См http://arduino.cc/en/Reference). Вы можете создать свои, более удобные лично Вам. Далее идет пример функции “add” (добавить), которая складывает два числа и возвращает результат.

```
int add(int параметр1, int параметр2)
{
        int x;

        x = параметр1 + параметр2;

        return
}
(x);
```

Ваши функции могут принимать и возвращать значениях "Параметров", как это сделано здесь. И конечно вы можете придумать что либо свое.

Если вы будете передавать параметры /к/ вашей функции, опишите и их типы в скобках, после имени функции. Если вы не будете передавать никаких параметров, вставьте скобки пустыми (), после имени функции.

Если вам нужно вернуть (получить) что либо /от/ вашей функции, поставьте тип возвращаемого значения перед именем функции. Тогда, когда ваша функция будет готова вернуть значение, вам останется положить его в return(). Если возвращать ничего не нужно, оставьте "void" перед именем функции (так же, как это делается в setup() и loop()).

Когда вы пишете свои собственные функции, старайтесь сделать код простым и аккуратным, для того чтобы упростить его повторное использование.





Откройте в Arduino IDE // Файл > Примеры > MaxKit.ru > lesson_11

На заметку:

```
char notes[] = "cdfda ag cdfdg gf ";

char names[] = {'c','d','e','f','g','a','b','C'};
```



До сих пор мы работали только с числовыми данными, но Arduino может работать и с буквами и с текстом. Символ, буква, цифра, и т.д., каждый имеет свой собственный тип называемый "char". Строка символов называется "string". Если у вас есть массив символов, он должен быть заключен в двойные кавычки, или же в виде списка в одиночные.

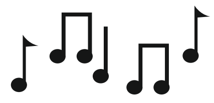
```
tone(pin, frequency, duration);

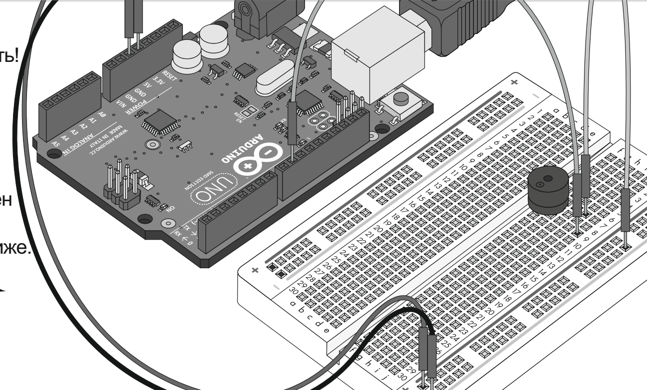
noTone(pin9);
```

Одной из многих, полезных команд Arduino является функция tone(). Эта функция выводит в порт "ноту", т.е. определенную частоту, что делает ее идеальной для работы с зуммерами и динамиками. Если вы даете паузу, после ноты (в милли сек.), она играет ноту и останавливается. Если паузу не сделать, нота будет звучать вечно. Хотя остановить ее можно и так noTone().

Что вы должны увидеть:

Что же вы должны увидеть? Ничего, но вы должны услышать! Ваша пищалка будет играть "Twinkle, Twinkle Little Star", или что то подобное. Если этого не происходит, проверьте правильность всех соединений, проверьте загружен ли код в плату Arduino. Или воспользуйтесь подсказками ниже.





Возможные трудности:

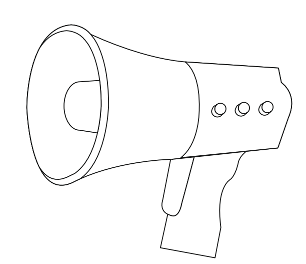
Нет звука
Учитывая размер и форму пищалки, легко промахнуться мимо нужного отверстия в плате. Попробуйте еще раз проверить его размещение.

Все равно не работает, не пойму почему
Попробуйте вытянуть пищалку из платы и снова воткнуть, и перезагрузите программу в Arduino.

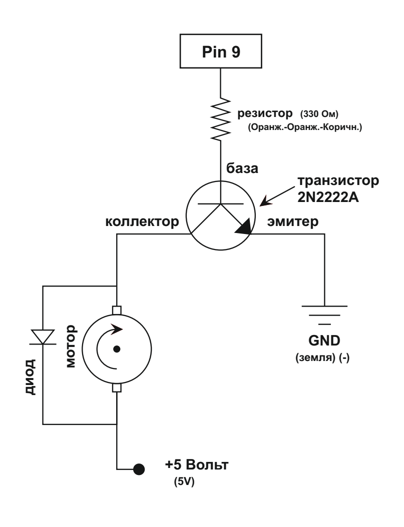
Устали от Twinkle Twinkle Little Starir
Код написан так, что вы легко можете добавить свои собственные мелодии.

Применение в жизни:

Многие мегафоны используют подобный метод усиления сигнала, хотя качество звука оставляет желать лучшего.



ОПЫТ 12



12 Крутись Мотор

Помните, ранее вы изучили серводвигатель? Теперь мы познакомимся с электродвигателем. Для это потребуется транзистор, который может коммутировать (переключать) больший ток, чем может себе позволить Arduino. Используя транзистор, нужно убедиться, что его параметры подходят для выбранного оборудования. Транзистор, который мы используем для этой рассчитан на максимальное напряжение 40 Вольт и 200 миллиампер, - идеально подходит для для нашего игрушечного двигателя!



Когда вы собираете схему, будьте осторожны и внимательны, не перепутайте транзистор с датчиком температуры, они очень похожи.

детали:

- Транзистор 2N2222A x1
- Диод 1N4148 x1
- Мотор x1
- Провод x6
- 330 Ом Резистор x1

2N2 222A A18

перед

зад

Опыт 12 : Крутись Мотор

Компоненты	Обозначение		
Транзистор 2N222	P2N2 222A A18		a1 - a2 - a3
Диод 1N4148			b7 - b11
Мотор			e7 - e11
Резистор 330 Ом			e2 - g2
Провод перемычка			e1 - -
Провод перемычка		Pin 9	j2
Провод перемычка			a7 - +
Провод перемычка			e3 - d11
Провод перемычка		5V	+
Провод перемычка		GND	-

Промежуточные итоги:

К этому моменту у вас, вероятно, уже начинают появляться собственные идеи и мысли, по поводу новых проектов, которые можно собрать, или вы уже решаете реально сложную задачу.
Несколько советов по дальнейшим действиям. Большинство программ, которые вы напишите, будут состоять из действий, выполняемых по кругу (loop), плюс еще несколько подобных пунктов ниже:

1. Выполнить какой-то ввод данных
2. Сделать некоторые расчеты, или сравнения
3. Получить, или извлечь полученный результат
4. Повторить действия, или нет!

Мы уже показали вам, как использовать множество различных датчиков и устройств для ввода данных, осталось еще несколько не пройденных опытов. Не стесняйтесь использовать чужие примеры из «Open Source» (для всех) в своих скетчах.
Собирайте кусочки из разных программ вместе, открывайте чужие скетчи в дополнительных окнах, и копируйте кусочки кода из одного окна, в другое. Так вы научитесь многим приемам программирования.
Используйте константы для номеров портов, создавайте функции в программах, они облегчают повторное использование кусков кода.

Если вам нужна помощь, отправляйтесь в соответствующий интернет форум с данной темой, там вы можете задать свой вопрос не стесняясь.
Попробуйте зайти на форум Arduino по адресу:
arduino.cc/forum, или другой, соответствующий вашим предпочтениям.
Когда вы будете готовы перейти к более сложным темам, загляните на страницы Arduino в учебники по адресу arduino.cc/en/Tutorial.
Много интересных статей можно найти на сайте http://arduinokit.ru

Наконец, если вы думаете что создали что-то действительно классное, рассмотрите вопрос о том, чтобы познакомить ваш проект с интернет сообществом, чтобы весь мир узнал гения создавшего что-то стоящее.
И не забудьте, дайте знать нам, мы разместим его на нашем сайте!
http://arduinokit.ru e-mail: **info@arduinokit.ru**



Код Arduino:



Откройте в Arduino IDE // Файл > Примеры > MaxKit.ru > lesson_12

На заметку:

```
while (Serial.available() > 0)
```



Последовательный порт Arduino может использоваться как для получения, так и для передачи данных. Поскольку данные могут прийти в любое момент, сначала они поступают в "хранилище", или Буфер обмена Arduino и хранятся там до тех пор, пока вы не будете готовы ими воспользоваться. Команда Serial.available() возвращает количество символов, которое порт получил, но вы ими еще не воспользовались. Ноль означает, данные отсутствуют.

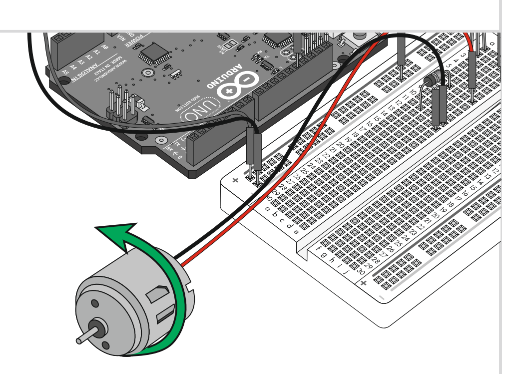
```
speed = Serial.parseInt();
```



Если в порт поступили данные и они находятся в буфере, есть несколько способов, чтобы извлечь их оттуда. Один из таких способов - использовать команду Serial.parseInt(), она производит поиск очередного целого числа в буфере последовательного порта, и если вы введете "1" "0" "0" в порт, эта функция вернет число 100.

Что вы должны увидеть:

Двигатель постоянного тока должен вращаться, если все компоненты цепи собраны правильно.
Если этого не происходит, проверьте загружен ли код в Arduino, а также ознакомьтесь с разделом поиска и устранения неисправностей ниже.



Возможные трудности:

Мотор не вращается
Если вы использовали свой транзистор, отличный от 2N2222, но подходящий по параметрам, проверьте соответствие его ножек по документации (возможен обратный порядок).

Все еще не повезло
Проверьте работает ли двигатель от 5 Вольт, и хватает ли для него мощности источника.

Все еще не работает
Иногда Arduino может отключаться от компьютера. Попробуйте отсоединить от нее USB разъем и затем снова подсоединить.

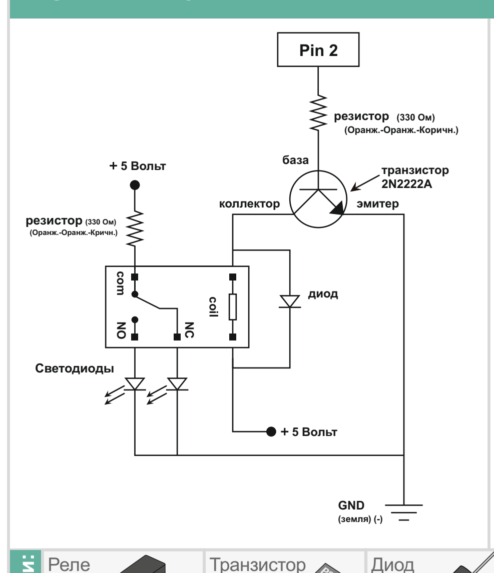
Применение в жизни:

Дистанционно управляемые автомобили используют двигатели постоянного тока, для поворота колес и для движения вперед, назад.



ОПЫТ 13

13



Реле

В этом опыте мы будем управлять реле, и для этого попробуем воспользоваться полученными знаниями из предыдущих 12 уроков.
Реле- это электрически управляемый, механический переключатель. Внутри этого простенького на первый взгляд, пластмассового корпуса, находится мощный электромагнит. И когда он получает заряд энергии, происходит срабатывание контактной группы, замыкая или размыкая цепь питания нагрузки. Из этого опыта вы узнаете, как Arduino управляет реле.

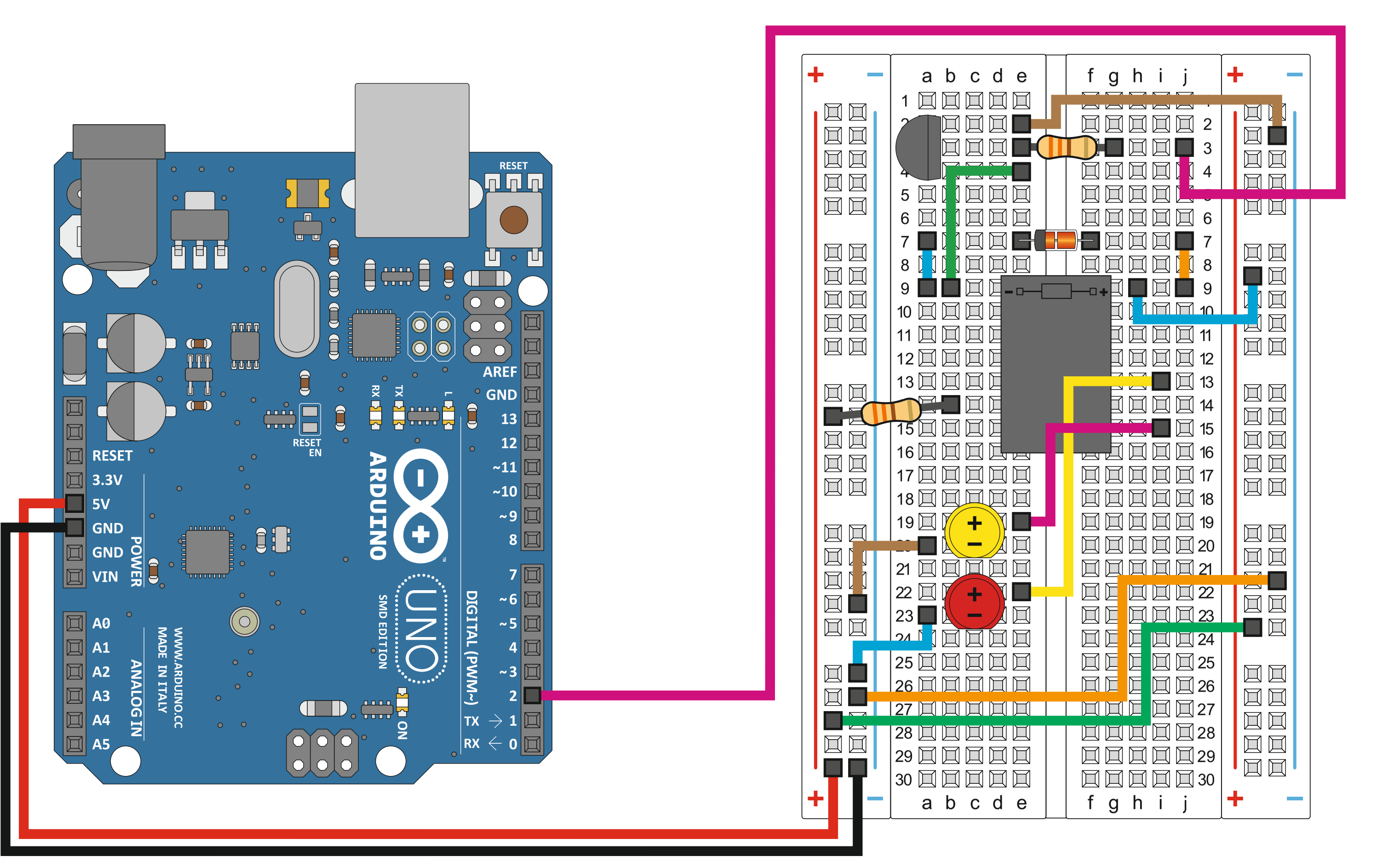


Возможно, в ваш набор вместо реле включен “электронный модуль реле”, его обозначение выводов должно совпадать с выводами реле. Когда реле выключено, общий контакт "COM" подключен к контакту "NC". Когда же реле сработает “COM” соединится с контактом "NO"

детали:

Реле		x1
Транзистор	2N2222A	x1
Диод	1N4148	x1
330 Ом Резистор		x2
Св. диод		x2
Провод		x14

Опыт 13: Реле



ARDUINO
UNO
SMD EDITION
DIGITAL (PWM~)
POWER
ANALOG IN
WWW.ARDUINO.CC
MADE IN ITALY
RESET
AREF
GND
3.3V
5V
VIN
A0
A1
A2
A3
A4
A5
ON
TX
RX

Компоненты	Обозначение			Компоненты	Обозначение		
Реле			e9 - f9, e15 - f15	Провод перемычка			- - e19
Транзистор 2N222	P2N2 222A A18		a2 - a3 - a4	Провод перемычка			+ - -
Светодиод	+ -		c19 (+) - c20 (-)	Провод перемычка			a23 - -
Светодиод	+ -		c22 (+) - c23 (-)	Провод перемычка			a20 - -
Диод 1N4148			b7 - b11	Провод перемычка			a7 - a9
Резистор 330 Ом			e3 - g3	Провод перемычка			e4 - e9
Резистор 330 Ом			e2 - g2	Провод перемычка		5V	+
Провод перемычка			e2 - -	Провод перемычка		GND	-
Провод перемычка		Pin 2	j3				
Провод перемычка			j7 - j9				
Провод перемычка			h9 - +				
Провод перемычка			i13 - e22				
Провод перемычка			i15 - e19				

13

Код Arduino:



Откройте в Arduino IDE // Файл > Примеры > MaxKit.ru > lesson_13

На заметку:



`digitalWrite(relayPin, HIGH);`



Когда мы "включаем" транзистор, он подключает цепь питания к обмотке катушки реле, контакты реле переключаются и соединяют общий контакт COM (common) с нормально разомкнутым контактом NO (Normally Open). И все что будет подключено к этой контактной группе - включится, так как цепь замкнется. У нас это светодиоды, но может быть что угодно.

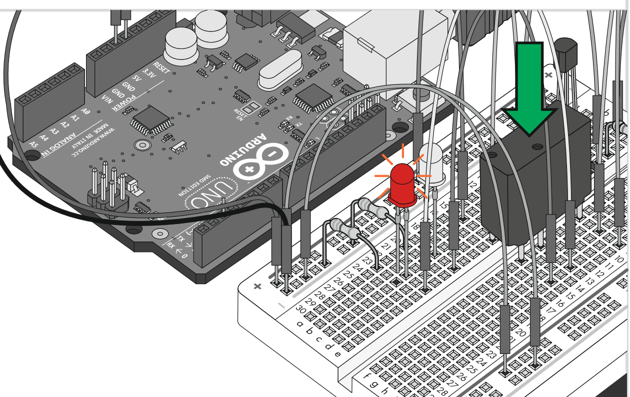
`digitalWrite(relayPin, LOW);`



У реле есть дополнительный, нормально замкнутый контакт NC (Normally Closed), когда реле выключено, он соединен с общим COM. Вы можете использовать любой, либо NO (нормально разомкнутый), либо (нормально сомкнутый) NC, в зависимости от того хотите ли вы что-то включить или включить. А можно переключать между двумя (как у нас).

Что вы должны увидеть:

Вы должны услышать щелчки переключающегося реле, а также увидеть два светодиода, попеременно загорающимися с секундным интервалом. Если этого нет, - проверьте правильно ли вы собрали схему, и загружен ли код в Arduino. Кроме того, смотрите советы по устранению неполадок.



Возможные трудности:

Светодиоды не светятся
Дважды проверьте правильность установки светодиодов, - длинный вывод является плюсовым контактом..

Не слышны щелчки реле
Проверьте правильность подключения реле и транзистора.

Срабатывает через раз
Проверьте надежность подключения реле. У реле, если это не электронный модуль, очень короткие выводы, попробуйте слегка придавить его в макетную плату.

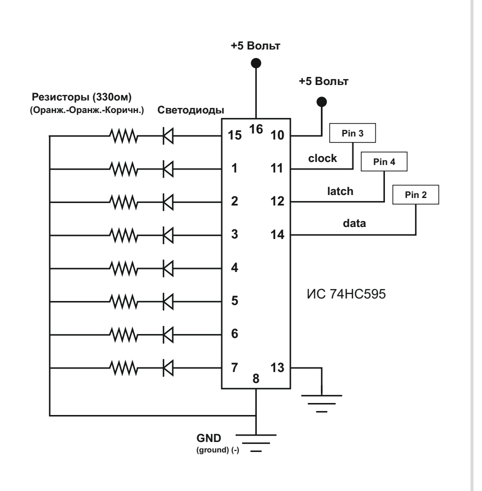
Применение в жизни:

Гаражные ворота - едут в одну сторону, затем в другую, и если прислушаться можно расслышать характерные для реле щелчки.

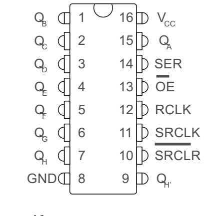
ОПЫТ #14

14

Регистр Сдвига



Наконец мы дошли до Интегральных Схем (ICs). В этом опыте вы узнаете об использовании ИС Регистра Сдвига и об SPI (Serial Periphera Interface). Последовательный периферийный Интерфейс ISP служит для передачи данных.
Один Регистр Сдвига добавит вашей Arduino дополнительные 8 выходов, заняв всего три. Схема текущего опыта позволяет управлять восемью светодиодами, используя лишь 3 порта Arduino.



Ключом является небольшая выемка на одном из торцов корпуса

ДЕТАЛИ:

ИС x1

Светодиод x8

Резистор 330 Ом x8

Провод x19

Слегка подогнуть

Опыт 14: Регистр Сдвига (Shift Register)

Компоненты:	Обозначение:			Компоненты:	Обозначение:		
ИС			e5 e6 e7 e8 e9 e10 e11 e12 / f5 f6 f7 f8 f9 f10 f11 f12	Провод перемычка			- - -
Светодиод 5мм	+ -		c14 + c15 -	Провод перемычка			j5 - +
Светодиод 5мм	+ -		c17 + c18 -	Провод перемычка			j6 - a14
Светодиод 5мм	+ -		c20 + c21 -	Провод перемычка		Pin 2	j7
Светодиод 5мм	+ -		c23 + c24 -	Провод перемычка			j8 - -
Светодиод 5мм	+ -		h14 + h15 -	Провод перемычка		Pin 4	j9
Светодиод 5мм	+ -		h17 + h18 -	Провод перемычка		Pin 3	j10
Светодиод 5мм	+ -		h20 + h21 -	Провод перемычка			j11 - +
Светодиод 5мм	+ -		h23 + h24 -	Провод перемычка			f14 - a8
Резистор 300 ом			- - a15	Провод перемычка			f17 - a9
Резистор 300 ом			- - a18	Провод перемычка			f20 - a10
Резистор 300 ом			- - a21	Провод перемычка			f23 - a11
Резистор 300 ом			- - a24	Провод перемычка			a23 - a7
Резистор 300 ом			j15 - -	Провод перемычка			a20 - a6
Резистор 300 ом			j18 - -	Провод перемычка			a17 - a5
Резистор 300 ом			j21 - -	Провод перемычка			a14 - j6
Резистор 300 ом			j24 - -	Провод перемычка		5V	+
Провод перемычка			+ - +	Провод перемычка		GND	-



Код Arduino:



Откройте в Arduino IDE // Файл > Примеры > MaxKit.ru > lesson_14

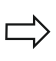
На заметку:

`shiftOut(datapin, clockpin, MSBFIRST, data);`



Ваше общение с Регистром Сдвига, и со многими другими элементами, происходит через специальный интерфейс “SPI” (последовательный периферийный интерфейс), линии передачи и синхронизации. Эти линии нужны для того, чтобы перемещать данные “data” из, или в Arduino на высокой скорости. Параметр MSBFIRST определяет порядок, в котором отправляются отдельные биты, мы посылаем самый старший бит первым.

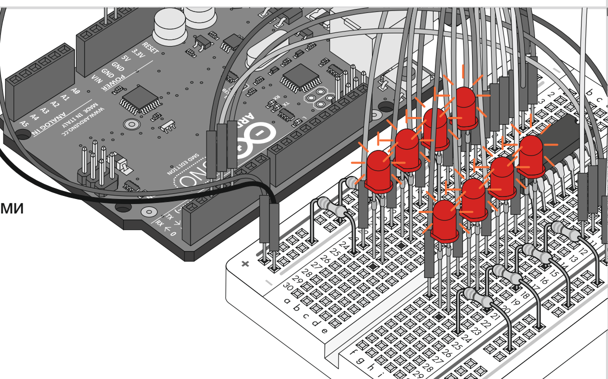
`bitWrite(data,desiredPin,desiredState);`



Бит занимает крошечный кусочек в памяти компьютера. Каждый такой бит может хранить в себе значение либо единицы "1", либо нуля "0". Большие числа хранятся как массивы бит. Иногда нам необходимо манипулировать значениями отдельных бит, сделать их либо 1, либо 0, как сейчас, когда мы посылаем их в Регистр Сдвига, чтобы включить или выключить светодиоды. У Arduino, для упрощения программы есть специальная команда bitWrite().

Что вы должны увидеть:

Вы должны увидеть различные световые эффекты на LED’s светодиодах, аналогично опыту №4, но с использованием ИС Регистра Сдвига.
Убедитесь в правильности сборки схемы. Проверьте загрузку кода в Arduino, или воспользуйтесь советами по устранению неполадок ниже.



Возможные трудности:

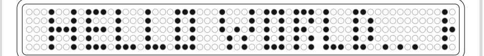
Индикатор питания Arduino гаснет
Такое иногда случается, если вставить ИС ключом не в ту сторону. Если быстро поправить ничего не сломается.

Работает не совсем правильно
Звучит как заезжанная пластинка, но возможно перепутаны местами провода.

Вам не понравилось
Пишите нам на e-mail, если схема показалась вам слишком сложной. Она одновременно и простая и сложная. Мы хотим узнать о ваших проблемах и пожеланиях: **info@maxkit.ru**

Применение в жизни:

Регистр сдвига обязательно используется в световом табло с бегущей строкой, которое состоит из огромного количества светодиодов.



Дополнительные сведения



Заходите к нам на сайт:

Эта книга - только начало ваших исследований в мире электроники, радиотехники и программирования. На нашем сайте вы найдете еще много интересного из различных областей, так или иначе связанного с компьютерами - веб дизайн, программирование, системное администрирование. И конечно огромное количество интересных статей и проектов, связанных с электроникой и робототехникой.

maxkit.ru
arduinokit.ru

ДЛЯ ЗАМЕТОК:

MaxKit.Ru

